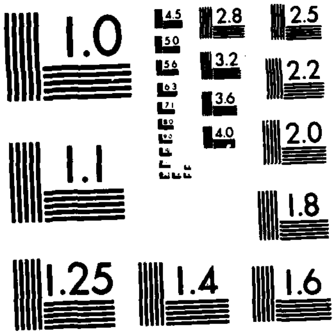
AD-A124 882   INVESTIGATION OF A THIRD ORDER BARO-DAMPED VERTICAL   1/2
CHANNEL OF INS(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING A RIAZ
UNCLASSIFIED   DEC 82 AFIT/GE/EE/82D-55   F/G 17/7   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

1

AD A124882

DTIC FILE COPY

INVESTIGATION OF A THIRD ORDER BARO-DAMPED VERTICAL CHANNEL OF INS

THESIS

AFIT/GE/EE/82D-5

Asad Riaz
Sqd Ldr, PAF

This document has been approved for public release and sale; its distribution is unlimited.

DTIC
ELECTE
FEB 2 4 1983
S D
A

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

83 02 024 098

AFIT/GE/EE/82D-55

INVESTIGATION OF A THIRD ORDER BARO-DAMPED VERTICAL CHANNEL OF INS

THESIS

AFIT/GE/EE/82D-55

Asad Riaz
Sqd Ldr, PAF

DTIC
ELECTE
FEB 4 [illegible]
A

Approved for Public Release; Distribution Unlimited

AFIT/GE/EE/82D-55

INVESTIGATION OF A THIRD-ORDER BARO-DAMPED VERTICAL CHANNEL OF INS

Thesis

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

Accession For
NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Avail and/or
Dist Special
A

by
Asad Riaz, B. E.
Sqd Ldr PAF
Graduate Guidance and Control
December 1982

DTIC COPY INSPECTED 2

Approved for Public Release; Distribution Unlimited

Preface

The intent of this study is to investigate the performance of the vertical channel for a vehicle carrying out a TERCOM-update in light of disturbance and noise inputs.

This project was sponsored by the Navigation Branch of the Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio.

I wish to express my appreciation to my advisor, Lt Col R.M. Edwards, for his responsiveness, guidance, assistance, and untiring patience to answer my endless questions throughout the study, and for his valuable suggestions to make this report as complete as possible. Special thanks to Mr. Robert Spaulding from Aeronautical Systems Division (ASD/ENACN), who originally conceived of the idea for this thesis, and Dr. George Siouris and Dr. Peter S. Maybeck for their contributions and helpful suggestions.

Finally, I would like to thank my wife, Zebinda, and my daughter, Ayesha, for their support and understanding during the completion of this thesis.

Asad Riaz

(This thesis was typed by Sharon Gabriel)

Table of Contents

Page

Preface-- ii

List of Figures---------------------------------- v

List of Tables----------------------------------- vii

Abstract--- viii

I. Introduction---------------------------------- 1

- Background------------------------------------ 1
- Problem--------------------------------------- 4
- Objectives------------------------------------ 4
- Approach-------------------------------------- 5
- Overview-------------------------------------- 7

II. Performance Assessment of Vertical Channel------- 8

- Selection of Model---------------------------- 8
- Simplified Model------------------------------ 10
- Addition of Uncertainties--------------------- 14
- Changed Model/Cost Function------------------- 17
- Mathematical Development---------------------- 23
- Typical Values of NSD/Correlation Parameters-------------------------------- 34

III. Program for Minimization of Cost----------------- 39

- Selection and Development of Routine---------- 39
- Validity Check of Program--------------------- 42
- Scaling and Techniques Used------------------- 42

IV. Error State Propagation and Simulation----------- 45

- The Truth Model------------------------------- 45
- Trajectory Selection-------------------------- 67
- Monte Carlo Simulation------------------------ 70

V. Results-- 72

- Basic Cost Function--------------------------- 72
- New Cost Function----------------------------- 74
- Sensitivity Study----------------------------- 80
- Simulation Results---------------------------- 82

VI. Conclusions and Recommendations----------------- 105

Table of Contents (Cont'd)

Page

Bibliography-- 108

Appendix A: Instability of Vertical Channel----------- 110

Appendix B: Minimization Algorithm Listing------------ 112

Appendix C: SOFE Input Data and User Routines--------- 118

Vita-- 145

List of Figures

Figure Page

1 Simplified Diagram of Vertical Channel Error Model with Disturbance Input----------- 11

2 Disturbance Profile------------------------- 13

3 Baro-Inertial Vertical Channel Error Model--- 15

4 Changed Model of Baro-Inertial Vertical Channel--------------------------------------- 20

5 Transfer Function Block from White Noise w_{a1} to Output-------------------------- 26

6 Final Transfer Function for Noise Source w_{a2} to Output------------------------- 28

7 Transfer Function Block from Noise Source w_{b1} to Output------------------------- 30

8 Final Transfer Function for Noise Source w_{b2}----------------------------------- 31

9 Flow Diagram of Search Routine-------------- 41

10 Upper 9x9 Partition of $\underline{F}(t)$----------------- 52

11 Added Elements to Figure 10 of LN-15 Fundamental Matrix--------------------------- 53

12 Partition of Gyro-Error State Variables------ 54

13 Partitions of F-Matrix of Accelerometer, Altimeter and Gravity Disturbance State Variables------------------------------------ 55

14 Altitude Profile---------------------------- 68

15 Vertical Velocity Profile------------------- 69

16 Altitude Error (Classical)------------------ 87

17 Altitude Error (Improved)------------------- 88

18 Altitude Error (Combined)------------------- 89

19 Vertical Velocity Error (Classical)---------- 90

List of Figures (Cont'd)

Figure Page

20 Vertical Velocity Error (Improved) ----------- 91

21 Vertical Velocity (Combined) ----------------- 92

22 Barometric Error (Classical) ----------------- 93

23 Barometric Error (Improved) ------------------ 94

24 Barometric Error (Combined) ------------------ 95

25 Altitude Error - One Run (Classical) --------- 96

26 Vertical Velocity Error - One Run (Classical) ------------------------------------ 97

27 Barometric Error - One Run (Classical) ------- 98

28 Altitude Error - One Run (Improved) ---------- 99

29 Vertical Velocity Error - One Run (Improved) ------------------------------------- 100

30 Barometric Error - One Run (Improved) -------- 101

31 Altitude Error - One Run (Combined) ---------- 102

32 Vertical Velocity Error - One Run (Combined) ------------------------------------- 103

33 Barometric Error - One Run (Combined) -------- 104

List of Tables

Table		Page
II-1	Nominal Values of Noise Spectral Densities and Correlation Parameter	38
IV-1	Error Model State Variables	47
IV-2	Notation Used in Figure 10 to Figure 13	56
IV-3	Error Source Initial Values and Statistics	61
V-1	Optimized Gains (Basic Cost Function)	72
V-2	Optimized Gains (New Cost Function)	76
V-3	Contribution of White Noises to Mean Squared Altitude Error	79
V-4	Optimized Gains for Zero Disturbance	81
V-5	Sensitivity of Gains	82

Abstract

The optimization of the three gains of a third-order baro-inertial vertical channel has been formulated as a stochastic optimal control problem, with the objective of minimizing the mean squared altitude error due to the noise induced altitude error and a disturbance of known magnitude.

For a vehicle carrying out a TERCOM-update immediately following a vertical descent, and being subjected to a disturbance input to the vertical channel, optimum gains are presented and the performance is analyzed through a simulated flight in a Monte Carlo analysis. Performance comparisons between the optimized gains and the classical gains are also presented. The results show a significant performance improvement over the classical gains for a vehicle carrying out the TERCOM-update.

INVESTIGATION OF A THIRD-ORDER BARO-DAMPED VERTICAL CHANNEL OF INS

I. Introduction

Background

The development of highly accurate, self-contained inertial navigation systems (INS) has been one of the major engineering accomplishments of the past fifty years. In the simplest terms, an inertial navigation system is one which uses Newton's law of motions and a set of initial conditions to determine continuously the velocity, position and attitude of the vehicle in which it is contained. The first aircraft navigation systems were primarily two-channel systems that provided horizontal navigation data (Refs 1,2,3). Inertial navigators using three channels were introduced with the advent of the missile and space era. In addition, the value of inertially derived altitude and vertical velocity was recognized in aircraft and missile applications involving low-level flights and precision weapon delivery (Ref 4).

Recently, the vertical channel performance has become important for a different reason. Long flights require some navigation update to counter the long term INS drifts. For cruise missiles, a position update has been developed based on pattern recognition of the terrain altitude profile. To measure this altitude profile, a radar altimeter measures terrain clearance while a constant altitude flight path is maintained. The constant altitude flight path depends on the indicated altitude from the INS. Clearly, INS errors directly corrupt the TERCOM (Terrain Contour Mapping) data. For this reason, accurate altitude tracking by the INS is of critical importance during the data taking period of TERCOM.

If one analyzes the error behavior of a local-level inertial navigation system, one finds that, given the Schuler, the Foucault and the 24-hour oscillations, the horizontal axes (east, north) display a stable navigation error behavior, while the vertical channel is unstable (Ref 5); that is to say, the vertical velocity and position errors increase exponentially with the passage of time (Appendix A). This instability is due to the calculation of the gravity correction; an accelerometer measures all the accelerations to which the vehicle is subjected with the exception of acceleration due to gravity. When INS acceleration is estimated by adding measured specific force to gravity computed from a gravity model, an error feedback is established due to evaluating the gravity model with an imperfect position estimate. In the vertical channel, this feedback is positive; that is, a positive vertical position error creates a positive vertical acceleration error. The essentially unstable nature of such a vertical channel mechanization results in errors which grow exponentially with an approximate ten minute time constant (Ref 5). Thus, for a typical navigation flight, the vertical channel needs to be stabilized by some external altitude reference, usually barometric altimeter data. Unfortunately, as normally implemented, this method has one small drawback. The time constant associated with a barometric altimeter is very large, which means that through prolonged descent or turns, the vertical channel inherits an error which can persist for as long as two minutes. This error can degrade weapon delivery, especially upon reattack.

A classical approach to improving and stabilizing the vertical channel is to introduce external altitude information from, for example, a barometric altimeter. The baro-damped vertical mechanization has evolved to a "third-order" mechanization which feeds back two terms to the vertical acceleration calculation, and one term to the vertical velocity calculation. The basic difference between baro-altimeter and the INS altiutde is fed back to the velocity calculation with a gain of K_1. This difference is also fed back to the acceleration calculation with a proportional gain of K_2 and an integral gain of K_3. In this manner, stable vertical channel operation has been developed which has proven acceptable for many applications.

Unfortunately, the classical gains (K_1, K_2 and K_3) result in a sluggish response to low frequency baro-induced altitude disturbances which are encountered during prolonged descents as might precede a TERCOM-update. Recent research in vertical velocity improvements (Ref 4) suggests that vertical position estimates might be similarly improved based on optimizing these gains.

Problem

This thesis addresses the task of optimizing the vertical position estimates of a baro-damped INS. Specifically, the third-order baro-damped system is treated to optimize the transient vertical performance by selecting proper gains (K_1, K_2 and K_3 for a third-order mechanization) during a TERCOM-type (Terrain Contour Mapping) update following specific disturbance profile to the vertical loop caused by vehicle maneuvers (horizontal or vertical turns).

Objectives

The objectives are to calculate optimal gains for the stated problem, investigate sensitivities of performance to these gains, and to validate the optimal gains in a Monte Carlo study.

Approach

The study will be based on a third-order mechanization error model of the vertical loop. In addition, the analysis will be restricted to the transient response of the vertical velocity and altitude following a series of specific maneuvers of the vehicle just before the TERCOM-update. The investigation will not include the steady-state analysis of the vertical loop; however, correlation between the steady-state following the transient behavior will be analyzed. In addition, theoretical complications and practical requirements will necessitate the imposition of certain assumptions:

1. It will be assumed that the vertical channel can be mechanized alone. This means that the coupling between the vertical and horizontal channels will be ignored. This coupling is not so in the real world environment, however; in this scope of study, it will not have a significant effect. For a full scale model, the coupling between the horizontal and vertical channels cannot be ignored.

2. Although a complete analysis of the system requires that both transient and steady-state behavior be categorized, for the purposes of analysis in this thesis, it will be assumed that the vertical loop of the INS closely follows the barometric altitude in steady-state. This assumption is true in the

practical world, provided there is a constant altitude flight for a period greater than the time constant of the baro-inertial vertical channel.

Until now, the present day mechanizations of the baro-aided vertical loop have been implemented by concentrating on the steady-state behavior of the inertial and barometric data. It is possible that shorter time constants and faster recovery time may yield more accurate instantaneous altitude and velocity at the expense of rather long term altitude errors such as those due to prolonged descents. This factor could significantly improve the performance of a vehicle carrying out a TERCOM-type update following immediately after a series of horizontal and vertical turns. The present day mechanizations have imbedded in them long time constants so that the INS altitude follows closely the barometric altitude in steady-state and neglects any variations in the latter due to standard setting or a scale factor error. However, in a prolonged descent/ascent, significant error develops in the barometric data with the INS closely following it. Thus, if a target needs to be attacked or if a TERCOM-update is required immediately following a prolonged ascent/descent, the vertical velocity and altitude will be in significant error.

The approach will be to simulate the vertical channel and model the error propagation in the presence of analytical models for the disturbance and to search for the optimum gains based on a cost function which concentrates on the TERCOM-type measurement update time-frame.

Overview

This thesis is presented in seven parts. First, Chapter I provides a background and the necessity for such an investigation. Chapter II discusses the model selection, including all uncertainties, and provides a cost function in light of the mathematical development presented. Chapter III details the minimization routine along with its verification and delineates pitfalls and solutions to possible problems which could be encountered during this process. In Chapter IV, the truth model and error state propagation of the LN-15 are presented for the Monte Carlo simulation. The trajectory for Monte Carlo simulation is also presented in this chapter. Chapter V presented the optimal gains and also the validation results from the Monte Carlo simulation. In Chapter VI, conclusions and recommendations of this thesis are presented. The Appendices contain the detailed description of the instability of the vertical channel and the computer listings for the minimization routine and user input routines for the Monte Carlo simulation.

II. Performance Assessment of Vertical Channel

Selection of Model

It is well known that, in the mechanization of a pure inertial navigation system, the calculation of altitude is unstable (Appendix A and Ref 5). Several methods have been proposed to stabilize the vertical channel. Various error models have been proposed depending on the actual application of the inertial system in the real world environment. In a conventional local-level system, the stabilization of the altitude is accomplished by correcting the vertical channel integrators with the difference between the inertial system and altimeter indication of the vertical position. Depending upon the complexity of the requirement, low-order to high-order mechanizations are used. Usually, however, a third-order mechanization is preferred for the reasons of optimum balance between performance and mathematical tractability. For this reason, a third-order mechanization of the vertical channel was chosen for the purposes of study in this thesis.

At present, the classical third-order mechanization is in widespread use. Efforts have been made toward improving the loop gains so as to obtain an equitable balance between the errors of the vertical velocity and altitude. Widnall and Sinha (Ref 4) formulated the selection of the three loop gains in the baro-inertial vertical channel as a stochastic optimal control problem, with the objective of minimizing the mean squared error of the indicated vertical velocity. With the optimum gains thus obtained, they showed an improvement of 30 percent over the classical set of gains in a simulated flight of an aircraft. A similar kind of study was carried out in this thesis with the objective of minimizing the mean squared error of the altitude at TERCOM-update. The error in altitude of the INS at TERCOM-update is of far greater importance than the error in vertical velocity. Error in vertical velocity is critical during a weapon delivery because this error greatly affects the miss distance of the weapon on a target. Since no weapon delivery is performed at TERCOM-update, it is logical to concentrate on minimizing the altitude error to protect against an incorrect or missed update of the navigation system. In addition to minimizing the mean squared altitude error, a non-stochastic disturbance from the baro-altimeter is modeled to account for the long term error introduced during a descent prior to TERCOM-update. Any gain selection should also treat this non-stochastic error source. By selecting gains to minimize altitude error due to this disturbance just prior to the update, the vertical channel performance can be optimized during the TERCOM-update. The basic error model formulated by Widnall and Sinha was used, and alterations were made according to the mission requirements, as will be shown in the succeeding paragraphs.

Simplified Model

Figure 1 shows the simplified version of the baro-inertial vertical channel error model. The set of equations describing this diagram is:

$$\delta\dot{h} = \delta V_Z - K_1 \, (\delta h - \delta D) \tag{1}$$

$$\delta\dot{V}_Z = (2g/R) \, \delta h - K_2 \, (\delta h - \delta D) - \delta\hat{a} \tag{2}$$

$$\delta\dot{\hat{a}} = K_3 \, (\delta h - \delta D) \tag{3}$$

where δh is the closed loop altitude error, δV_Z is the vertical velocity error, δD represents the disturbance input and is the variation of sensed altitude error from the true value, and $\delta\hat{a}$ is the vertical acceleration error estimate variable. The loop gains are given by K_1, K_2, K_3, and g is the magnitude of gravity computed as a function of indicated altiutde and latitude with R being the geocentric radius. The feedback of δh through $2g/R$ reflects changes in gravity with altitude, and it can be recognized as the cause of instability in the unaided

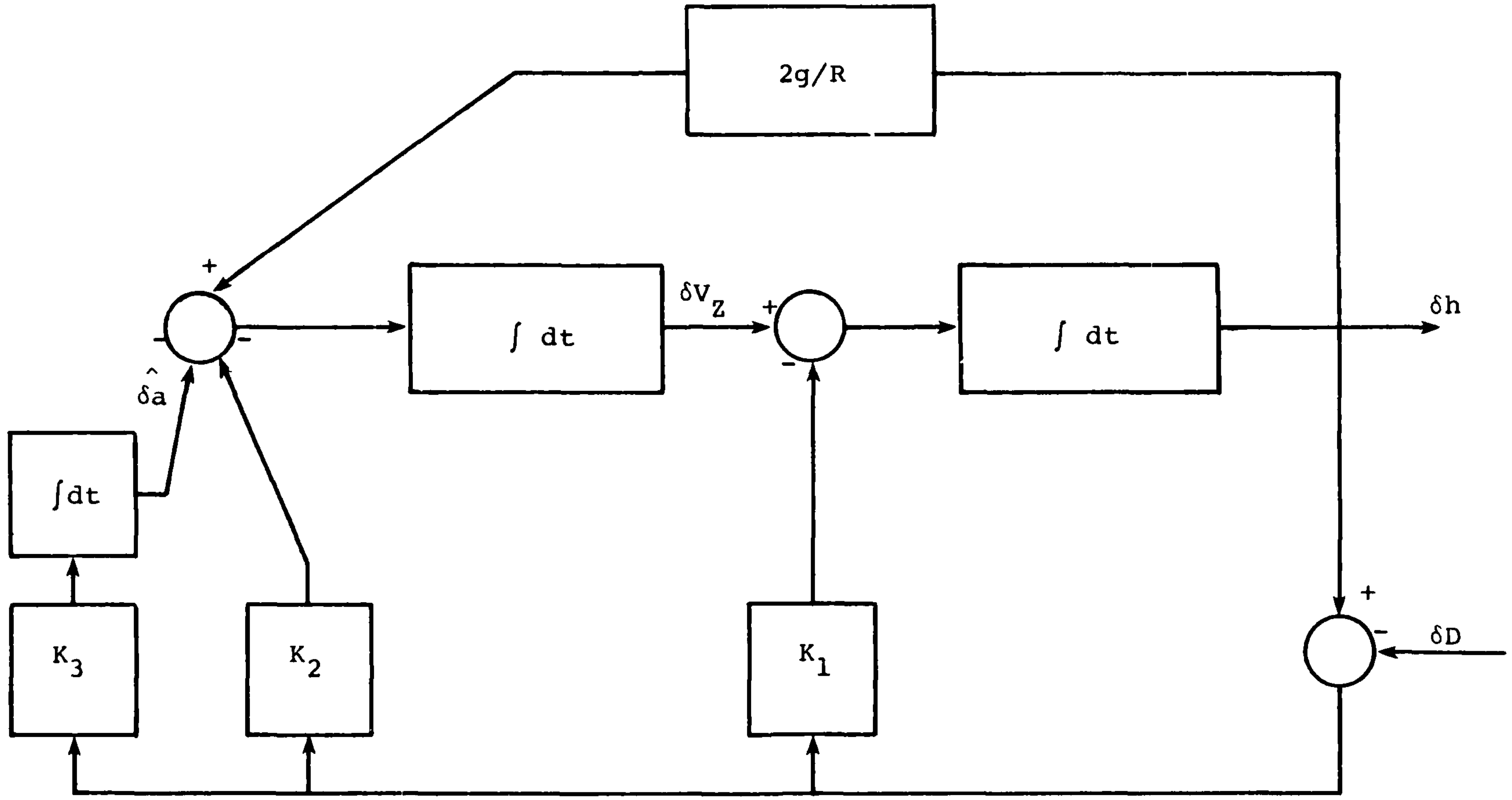


Figure 1. Simplified Diagram of Vertical Channel Error Model with Disturbance Input

inertial vertical channel. Although a simpler second-order damping system would result from setting K_3 equal to zero, the performance advantage of the more complex system is sufficient to warrant its use.

It will be assumed that the disturbance (δD) acts on the vertical channel for a time interval Δt_1 which begins at some time t_1 and terminates at time t_2 (Fig. 2). It will be further assumed that it is desired to minimize the altitude error over an interval Δt_2 which begins at time t_3 where $t_3 > t_2$ (Fig. 2).

In light of the above statements, the cost function to be minimized over Δt_2 is

$$J(\underline{K}) = \int_{t_3}^{t_4} (\delta h)^2 \, dt \quad ; \quad \underline{K} = K_1, K_2, K_3 \tag{4}$$

where $J(\underline{K})$ is the performance index which is an arbitrary mathematical expression designed to measure how well a system performs a particular task. Since both positive and negative values of the altitude error are equally undesirable, the measurement of the mean squared error of the altitude is an appropriate means of indicating how well the INS performs over the defined time interval. Another form of the cost function of Eq (4), although mathematically less desirable, would be to replace the square of the altitude error by simply its magnitude.

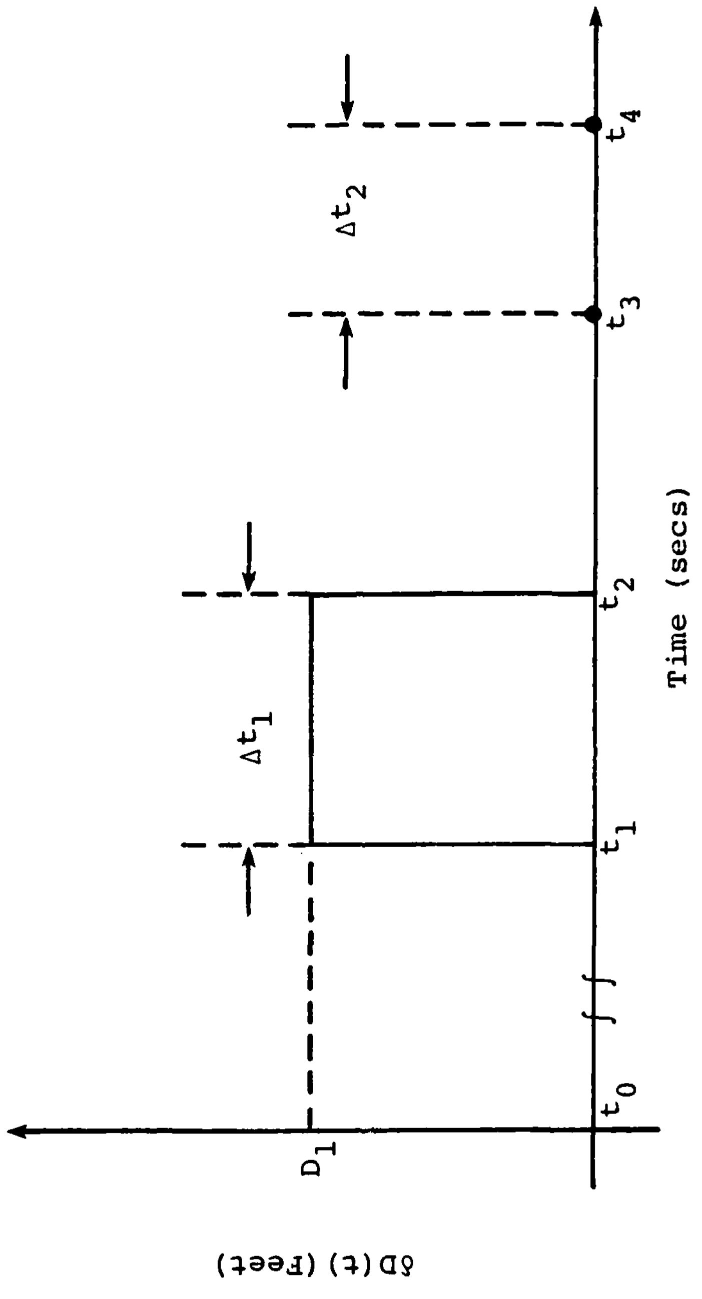


Figure 2. Disturbance Profile

Equation (4) is the basic cost function and was minimized through a search routine, the results/discussions of which are given in Chapter V.

Addition of Uncertainties

The diagram of Figure 1 is by no means a comprehensive depiction of all the errors associated with the vertical channel. Numerous other error sources associated with the vertical channel must be modeled to account for known error producing mechanisms. Although the various error sources have been modeled as white noises and random walks, there are better models than just these. It is only for the sake of mathematical tractability that the simpler models are preferred whenever possible. A comprehensive diagram is shown in Figure 3.

The feedback path $(2g/R)$ arises from the gravity calculation and has the effect of destabilizing the altitude (Appendix A). The error state δa is a random walk and it models the following (Ref 4):

1. Bias or slowly varying error in the vertical acceleration due to accelerometer bias.
2. Gravity anomoly.
3. Error in Coriolis terms.

The white noise w_{a2} feeding into the integrator provides the randcm walk for the error state δa . The

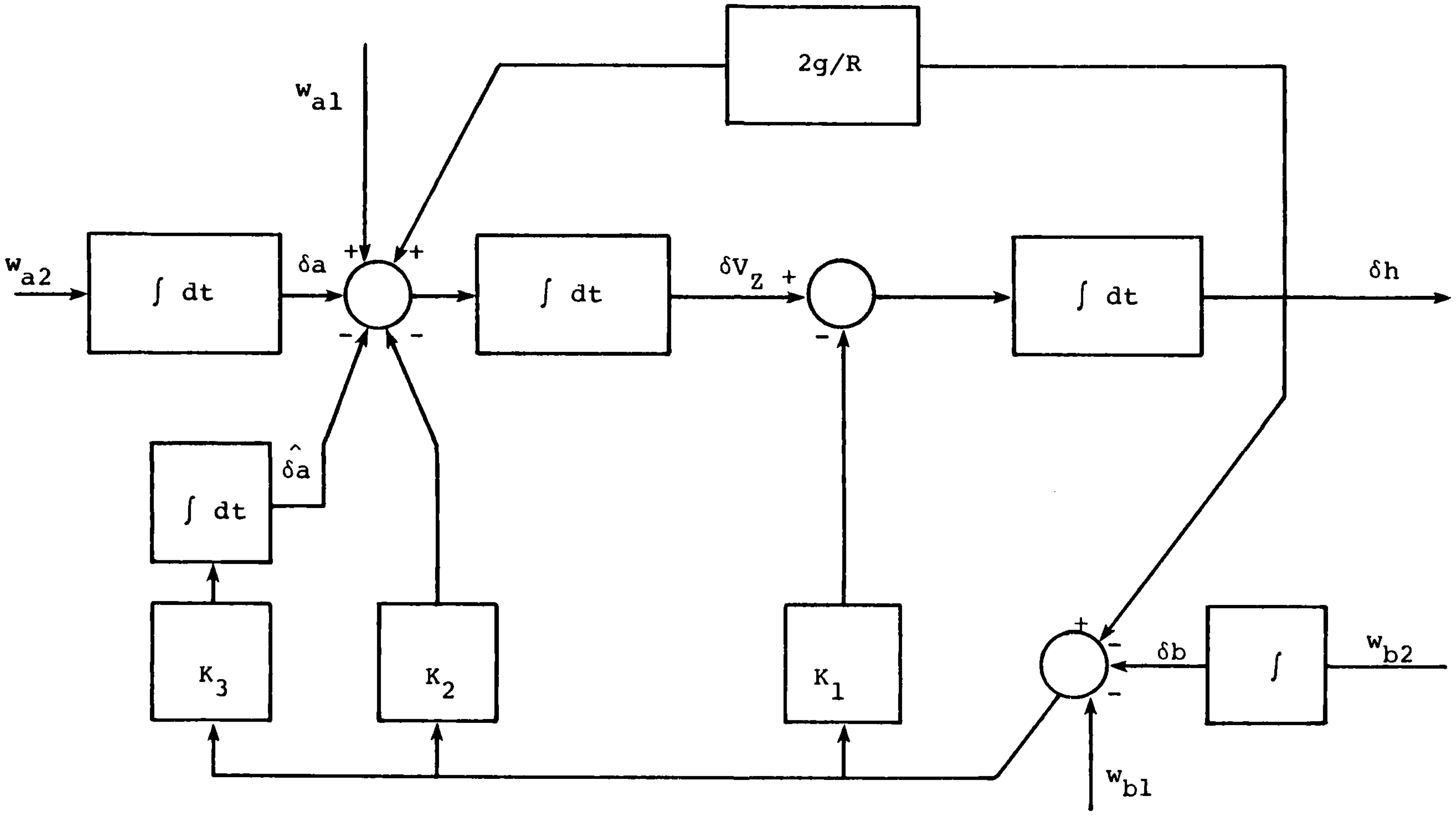


Figure 3. Baro-Inertial Vertical Channel Error Model

white noise w_{a1} into the summing junction models short correlation time accelerometer error, which could arise due to the vertical accelerometer scale factor error and input axis misalignment during the maneuver of the vehicle (Ref 4). The random white noise w_{b1} models any short correlation time altimeter error due to change in side-slip angle or in angle of attack during a maneuver (Ref 4). The white noise w_{b2} provides the random walk for the error state δb which represents the baro-altimeter error and is the sum of terms as follows:

$$\delta b = e_{p_o} + h\, e_{hsf} + c_{sp} V^2 - \tau_b V_Z + \delta D \tag{5}$$

where e_{p_o} is the altimeter error due to altimeter bias, e_{hsf} is the altimeter scale factor error, c_{sp} represents the static pressure measurement error and τ_b is the altimeter lag during ascents/descents. The additional term δD represents the disturbance input to the baro-altimeter, which is present only during the time interval Δt_2 as stated previously. It is assumed here that the disturbance input is uncorrelated with all the four white gaussian noises. The modeling details of the terms of Eq (5) are presented in Chapter IV under the Truth Model heading.

Changed Model/Cost Function

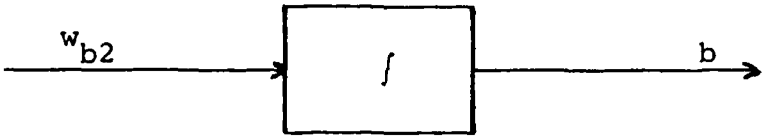
In Figure 3, the baro-altimeter error δb is modeled as a random walk with the white noise w_{b2} driving force on this. The random walk is the output of an integrator driven by white gaussian noise. Thus

$$\delta\dot{b}(t) = w_{b2}(t) \quad ; \quad \delta b(t_0) = 0 \tag{6}$$

where the white noise w_{b2} has zero mean and covariance dynamics.

$$\dot{P}_{\delta b \delta b}(t) = Q \tag{7}$$

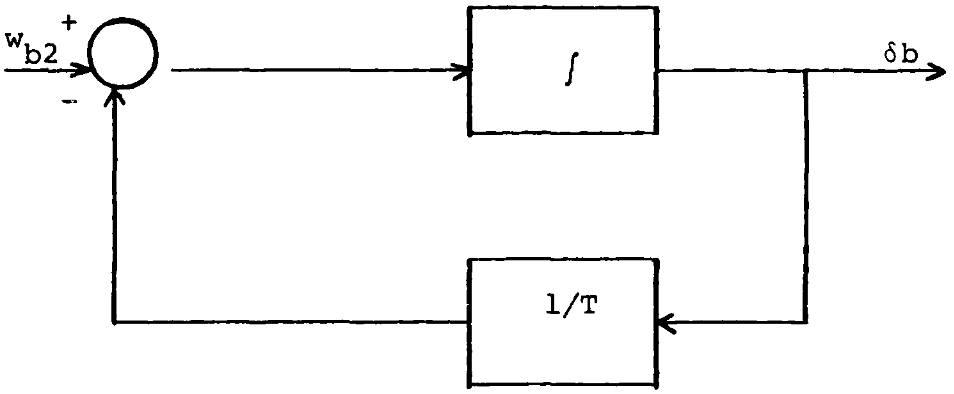
where Q is the strength of the white noise driving the integrator.



It can be seen from Eq (7) that the mean squared value grows linearly with time and is unbounded; i.e.,

$$E\{\delta b^2(t)\} = Q[t - t_o] \tag{8}$$

Since no term of Eq (5) grows unbounded with time, it is inappropriate to model δb as a random walk. For this reason, this model was deleted and instead δb was modeled as the output of a first-order lag driven by white gaussian noise as follows (Ref 8) (the selection of T is done in the next section):



This model produces an autocorrelation

$$\psi_{\delta b \delta b}(\tau) = E\{\delta b(t)\,\delta b(t+\tau)\} = \sigma^2 e^{-|\tau|/T} \tag{9}$$

i.e., of correlation time T and mean squared value σ^2 (with zero mean) (the selection of T and σ is done in the next section). Thus, the first order lag can be described as

$$\delta\dot{b}(t) = -(1/T)\,\delta b(t) + w(t) \tag{10}$$

where Q is the strength of the white noise and the mean squared value of the process is

$$E\{\delta b^2(t)\} = \frac{QT}{2} = \sigma^2 \tag{11}$$

where

$$Q = 2\sigma^2/T \tag{12}$$

It can be clearly seen from Eq (11) that the variance is constant.

The changed model of the baro-inertial vertical channel is depicted in Figure 4, where the baro-altimeter error is modeled as the output of a first-order lag driven by white gaussian noise w_{b2} .

In the presence of random inputs, as is the case here, it is more appropriate to take the expected value of the cost function denoted as $E\{\cdot\}$ herein. Recalling Eq (4), the cost function can be written as

$$J(\underline{K}) = E\{\int_{t_3}^{t_4} \delta h^2(t)\, dt\} \tag{13}$$

Define

$$\delta h(t) = \delta h_1(t) + \delta h_2(t) \tag{14}$$

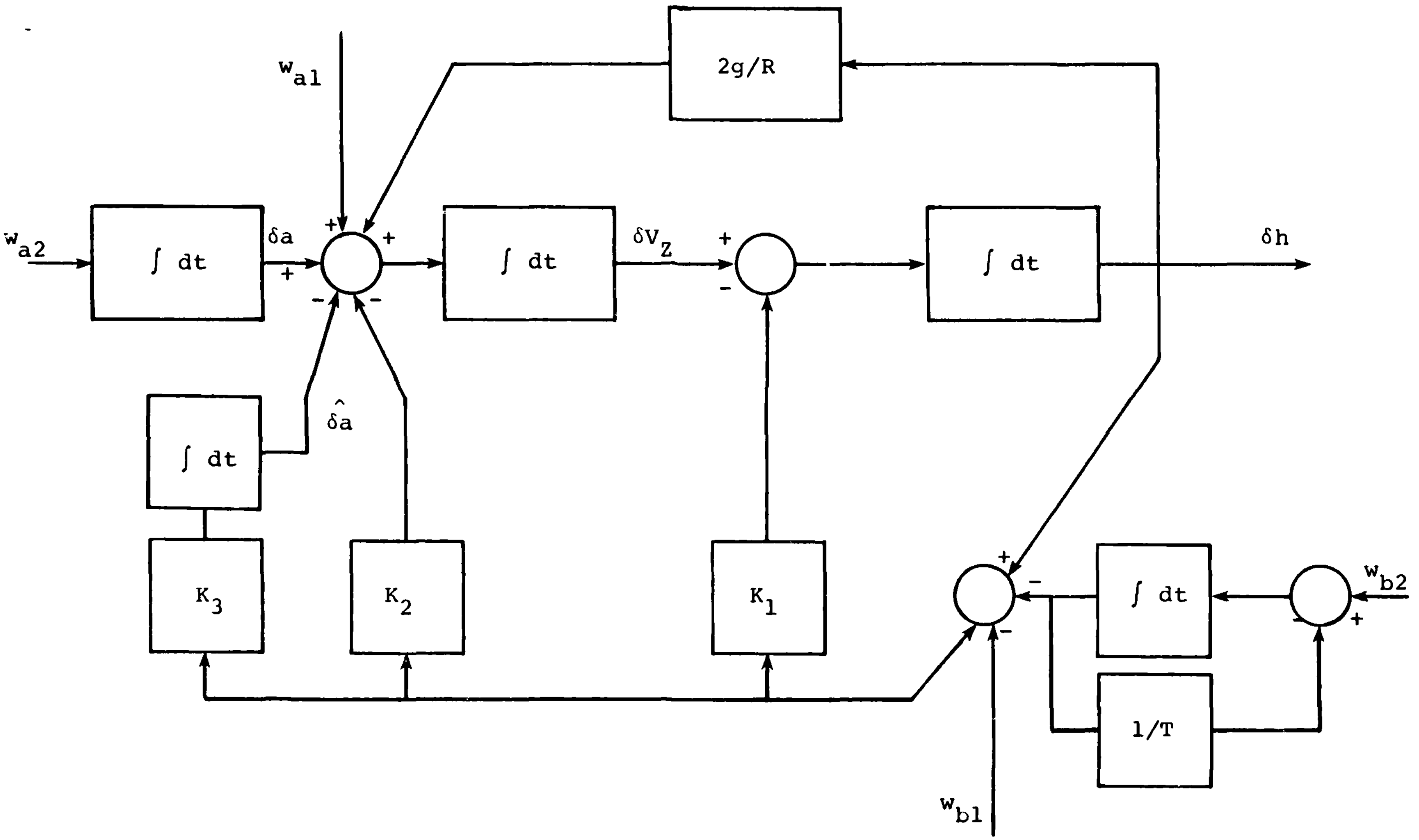


Figure 4. Changed Model of Baro-Inertial Vertical Channel

where $\delta h_1(t)$ and $\delta h_2(t)$ are the errors in altitude due to the deterministic (disturbance) and white gaussian noises (of mean zero), respectively. Taking the square of Eq (14) yields

$$\delta h^2(t) = \delta h_1^2(t) + 2\delta h_1(t)\delta h_2(t) + \delta h_2^2(t) \tag{14a}$$

Rewriting the right hand side of Eq (13)

$$E\{\int_{t_3}^{t_4} \delta h^2(t)dt\} = \int_{t_3}^{t_4} E\{\delta h^2(t)\}dt \tag{15}$$

Using Eq (14a),

$$\begin{aligned} E\{\int_{t_3}^{t_4} \delta h^2(t)dt\} &= \int_{t_3}^{t_4} E\{\delta h_1^2(t)dt\}dt \\ &+ 2\int_{t_3}^{t_4} E\{\delta h_1(t)\delta h_2(t)\}dt \\ &+ \int_{t_3}^{t_4} E\{\delta h_2^2(t)\}dt \end{aligned} \tag{16}$$

Since the disturbance is assumed independent of the white gaussian noises (as stated earlier), Eq (16) simplifies to

$$E\{\int_{t_3}^{t_4} \delta h^2(t)dt\} = \int_{t_3}^{t_4} E\{\delta h_1^2(t)\}dt$$

$$+ \int_{t_3}^{t_4} E\{\delta h_2^2(t)\}dt \tag{17}$$

By our definition of $\delta h_1(t)$ being the error due to a deterministic input, the expectation on $\delta h_1(t)$ can be removed

$$J(\underline{K}) = \int_{t_3}^{t_4} \delta h_1^2(t)dt + \int_{t_3}^{t_4} E\{\delta h_2^2(t)\}dt \tag{18}$$

If the mean square value of the altitude error due to zero-mean stochastic inputs remains constant over the TERCOM-update interval, i.e., Δt_2 (see Fig. 2), then Eq (18) can be finally written as

$$J(\underline{K}) = \int_{t_3}^{t_4} \delta h_1^2(t)dt + P_{\delta h}\,\Delta t \tag{19}$$

where $P_{\delta h}$ is the covariance (or mean squared value since the stochastic inputs are zero mean) of the altitude error (stochastic inputs) over the interval $\Delta t = (t_4 - t_3)$. The first term on the right hand side of Eq (19) is due to the deterministic input (as in Eq (4)) and the second one is due to the zero mean stochastic inputs. Equation (19) can be written as the sum of two cost functions

$$J(\underline{K}) = J_1(h) + J_2(h) \tag{19a}$$

where

$$J_1(\underline{K}) = \int_{t_3}^{t_4} \delta h_1^2(t)\,dt \tag{19b}$$

and

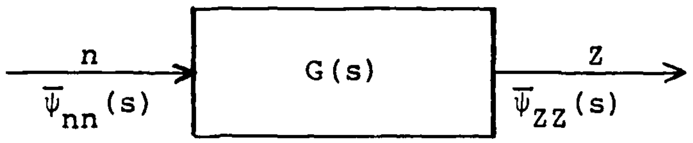
$$J_2(\underline{K}) = P_{\delta h}\,\Delta t \tag{19c}$$

Thus, the minimization of the cost function of Eq (19) will lead to the minimization of the altitude error and the disturbance over a time interval Δt_2 . The results of minimization of Eq (19) and discussions are presented in Chapter V under the New Cost Function heading.

Mathematical Development

As stated in the previous section, it is required to compute the mean squared altitude error as a function of input noise spectral densities and loop gains. This expression is required in Eq (19) to compute the overall cost function.

It is useful to express the power spectral density of a wide-sense stationary output of a system directly in terms of the power spectral density of the input and the description of the system itself (Ref 8).



From above, for a system of transfer function $G(s)$ and input and output power spectral densities of $\overline{\Psi}_{nn}(s)$ and $\overline{\Psi}_{ZZ}(s)$, respectively, it is true that (Ref 8)

$$\overline{\Psi}_{ZZ}(\omega) = G(-\omega)G(\omega)\overline{\Psi}_{nn}(\omega) \tag{20}$$

If input is a white gaussian noise of strength Q for all ω , then Eq (20) becomes

$$\overline{\Psi}_{ZZ}(\omega) = G(-\omega)G(\omega)Q \tag{21}$$

Similar to the lines of the development of Eq (21), the power spectral density of the altitude error can be computed from the following equation:

$$\overline{\Psi}_{\delta h}(\omega) = \sum_{i=1}^{4} G_i(j\omega)\ G_i(-j\omega)\ Q_i \tag{22}$$

or

$$\overline{\Psi}_{\delta h}(S) = \sum_{i=1}^{4} G_i(S)\ G_i(-S)\ Q_i \tag{22a}$$

where G_i is the transfer function from each of the independent white noise sources to the output (Fig. 3) and Q_i is the strength of the individual white noises of all the four sources. The mean squared value of the altitude error due to stochastic inputs is then the integral of the power spectral density

$$\overline{(\delta h_2)^2} = \sum_{i=1}^{4} \frac{Q_i}{2\Pi j} \int_{-j\infty}^{j\infty} G_i(S)\ G_i(-S)\ dS \tag{23}$$

To calculate the mean squared altitude error, we first need to calculate the trnasfer function from each of the individual white noise sources to the output. This simple expression results from the assumption that the white noise sources are uncorrelated and independent, since each error is from a different source and there is not reason to believe that they are correlated.

Figure 5 shows the various transfer function blocks between the white noise w_{a1} and δh .

The overall transfer function, for the first noise source w_{a1} , is then (from Fig. 5)

$$G_1(s) = \frac{\delta h}{w_{a1}} = \frac{S}{K_3 + (K_2 - C)S + S\ (S + K_1)} \tag{24}$$

or

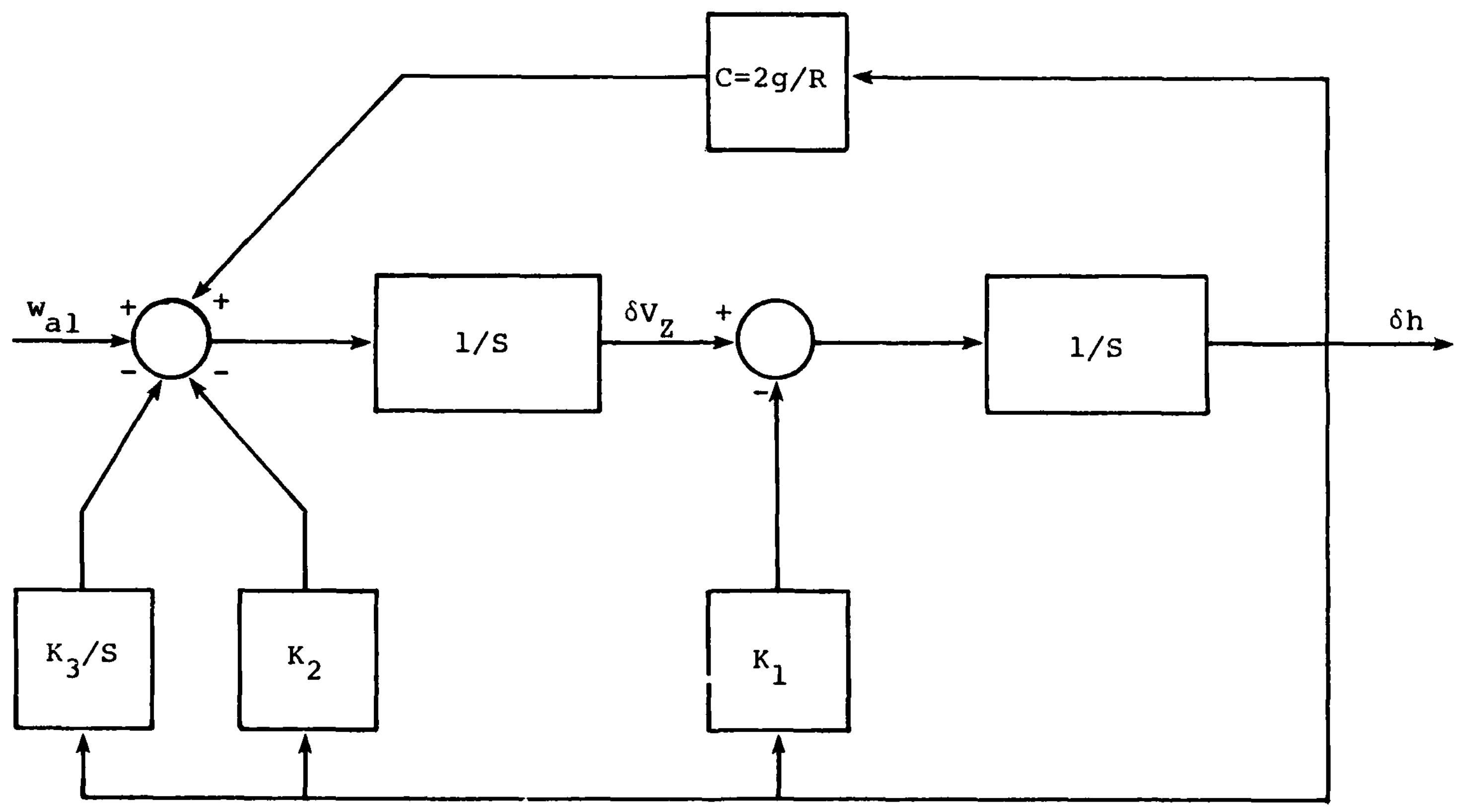


Figure 5. Transfer Function Block from White Noise w_{a1} to Output

$$G_1(S) = \frac{\delta h}{w_{a1}} = \frac{S}{S + K_1 S + (K_2 - C)S + K_3} \tag{25}$$

With Q_{a1} being the strength of white gaussian noise w_{a1} and using Eq (23) for $i=1$, we get

$$\overline{(\delta h_2)^2_1} = \frac{Q_{a1}}{2\Pi j} \int_{-j\infty}^{j\infty} G_1(S) G_1(-S)\, dS \tag{26}$$

Using the table of integrals (Ref 9), we get

$$\overline{(\delta h_2)^2_1} = \left\{\frac{1}{2\, K_1(K_2 - C) - K_3}\right\} Q_{a1} \tag{27}$$

For the second noise source w_{a2}, an additional integrator to Figure 5 is needed, which is shown in Figure 6. From Figure 6, the transfer function from white gaussian noise w_{a2} of strength Q_{a2} to the output is

$$G_2(S) = \frac{\delta h}{w_{a2}} = \frac{1}{S[S + K_1 S + (K_2 - C)S + K_3]} \tag{28}$$

As before, using Eq (23) for $i=2$ with Q_{a2} being the strength of w_{a2},

$$\overline{(\delta h_2)^2_2} = \frac{Q_{a2}}{2\Pi j} \int_{-j\infty}^{j\infty} G_2(S) G_2(-S)\, dS \tag{29}$$

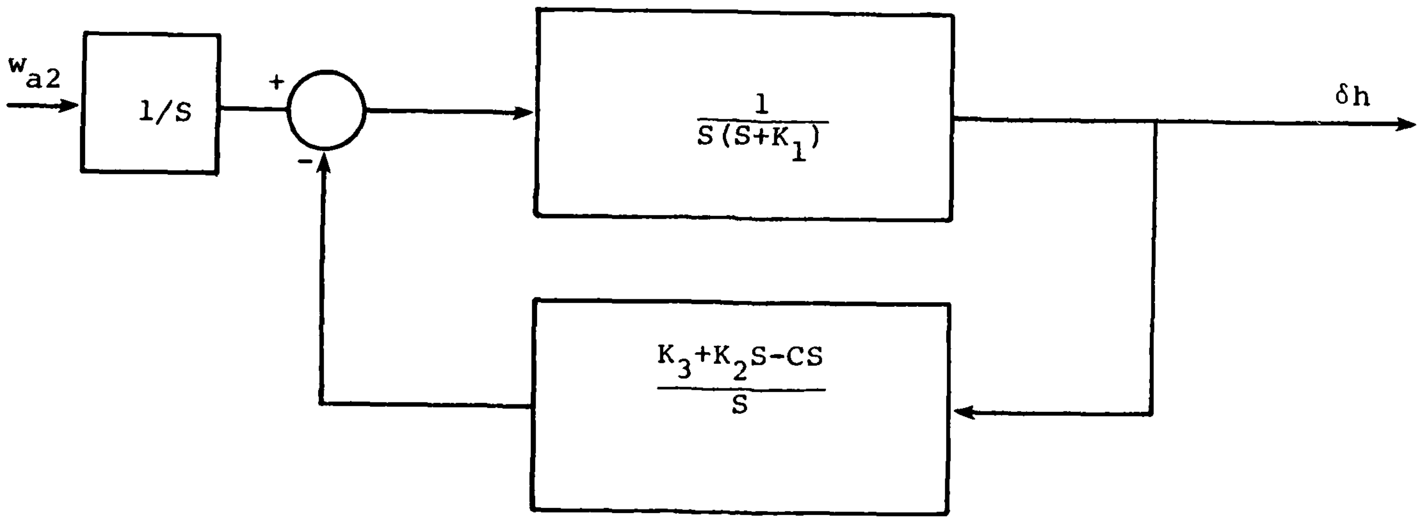


Figure 6. Final Transfer Function for Noise Source w_{a2} to Output

Using the table of integrals (Ref 9)

$$\overline{(\delta h_2)^2_2} = \{\frac{K_1}{2[K_1K_3(K_2 - C) - K_3^2]}\} \tag{30}$$

To calculate the transfer function for noise source w_{b1} to output, we proceed in a similar fashion as before. Figure 7 shows the block diagram from noise source w_{b1} to output. The overall transfer function for the third noise source w_{b1} is

$$G_3(S) = \frac{\delta h}{w_{b1}} = \frac{K_1S^2 + K_2S + K_3}{S^3 + K_1S^2 + (K_2 - C)S + K_3} \tag{31}$$

with Q_{b1} being the strength of w_{b1} and using Eq (23) for $i=3$.

$$\overline{(\delta h_2)^2_3} = \frac{Q_{b1}}{2\Pi j} \int_{-j\infty}^{j\infty} G_3(S)G_3(-S)\, dS \tag{32}$$

Using the integral tables (Ref 9)

$$\overline{(\delta h_2)^2_3} = \{\frac{K_1^2(K_2 - C) + K_2^2 - K_1K_3}{2[K_1(K_2 - C) - K_3]}\} Q_{b1} \tag{33}$$

Lastly, for the noise source w_{b2} , an addition of the first-order lag to Figure 7 gives the desired transfer function and is depicted in Figure 8. From Figure 8,

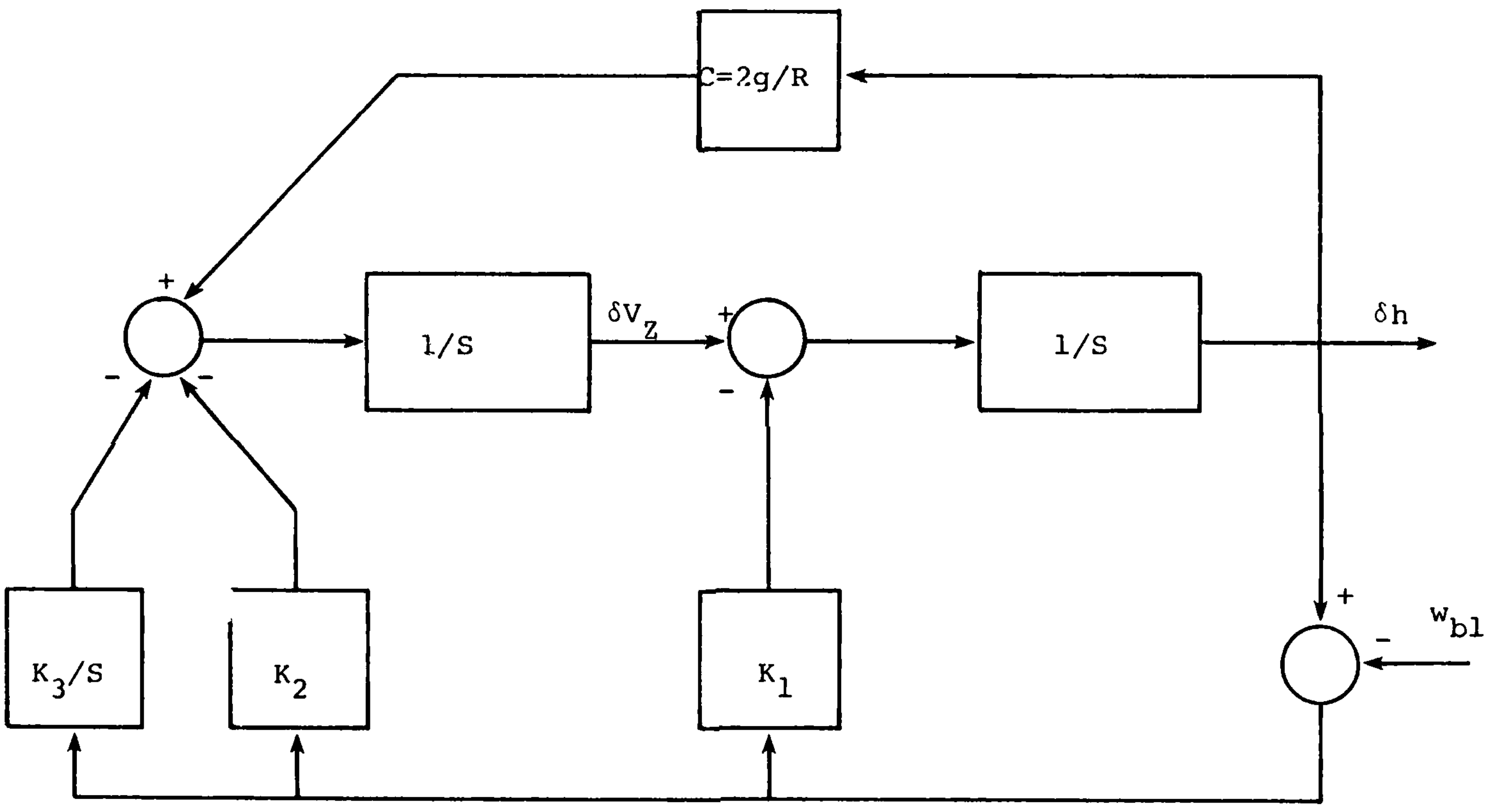


Figure 7. Transfer Function Block from Noise Source w_{b1} to Output

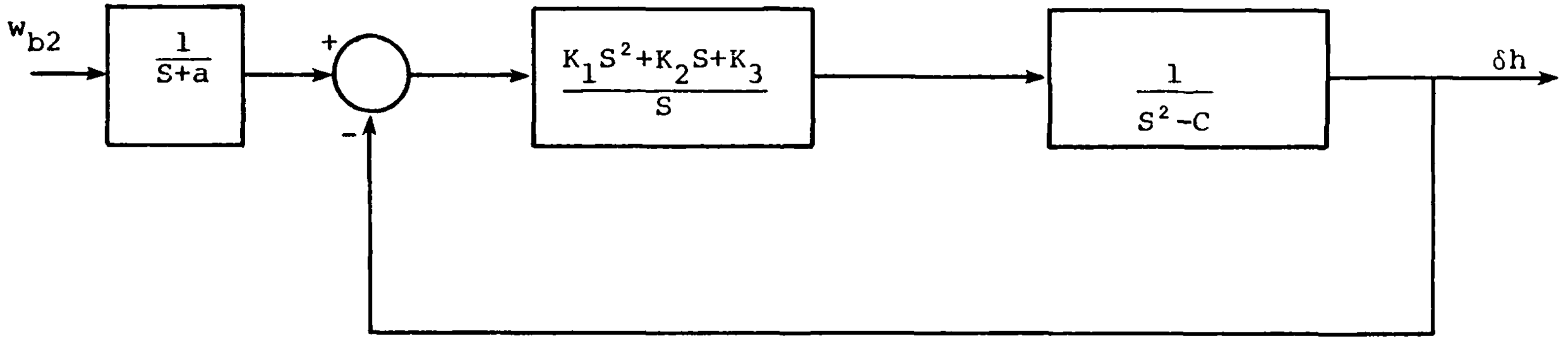


Figure 8. Final Transfer Function Block from Noise Source w_{b2} to Output

$$G_4(S) = \frac{\delta h}{w_{b2}} = \frac{K_1S^2 + K_2S + K_3}{(S + a)[S^3 + K_1S^2 + (K_2 - C)S + K_3]} \tag{34}$$

$$= \frac{K_1S^2 + K_2S + K_3}{S^4+(a+K_1)S^3 + (aK_1+K_2-C)S^2 + (aK_2-aC+K_3)S + aK_3} \tag{35}$$

With Q_{b2} being the strength of w_{b2} and using Eq (23) for i=4

$$\overline{(\delta h_2)^2_4} = \frac{Q_{b2}}{2\Pi j} \int_{-j\infty}^{j\infty} G_4(S)G_4(-S)\ dS \tag{36}$$

Using the integral tables (Ref 9), the solution to Eq (36) is

$$\overline{(\delta h_2)^2_4} = \frac{\begin{pmatrix} a^2[K_1^2(K_2-C)-K_1K_3+K_2^2] + aK_1K_2^2 \\ +K_3[K_1(K_2-C) - K_3] \end{pmatrix} Q_{b2}}{2a\begin{bmatrix} a^3\{K_1(K_2-C)-K_3\}+ a^2\{K_1(K_1(K_2-C)-K_3)\} \\ + a\{K_1K_2(K_2-C) - K_3(K_2-C)-CK_1(K_2-C)\} \\ + K_3\{K_1(K_2-C) - K_3\} \end{bmatrix}} \tag{37}$$

Combining Eqs (27), (30), (33) and (37) and substituting in Eq (23), the mean squared altitude error becomes

$$\overline{(\delta h_2)^2} = \frac{Q_{a1}}{2[K_1(K_2-C) - K_3]} + \frac{(K_1)Q_{a2}}{2[K_1K_3(K_2-C) - K_3^2]}$$

$$+ \frac{(K_1^2(K_2-C) + K_2^2 - K_1K_3)\, Q_{b1}}{2[K_1(K_2-C) - K_3]}$$

$$+ \frac{\left(\begin{matrix} a^2[K_1^2(K_2-C) - K_1K_3 + K_2^2] + aK_1K_2^2 \\ + \quad K_3[K_1(K_2-C) - K_3] \end{matrix}\right) Q_{b2}}{2a\left[\begin{matrix} a^3\{K_1(K_2-C)-K_3\}+a^2\{K_1(K_1(K_2-C)-K_3)\} \\ + a\{K_1K_2(K_2-C)-K_3(K_2-C)-CK_1(K_2-C)\} \\ + K_3\{K_1(K_2-C)-K_3\} \end{matrix}\right]} \tag{38}$$

Equation (38) is then the required mean-squared altitude error for use in Eq (19) to make the complete cost function.

One more aspect still remains untouched. Appropriate values for the strength of the white gaussian noises and the correlation parameter of the first-order lag are required. It is rather difficult to suggest values which provide a true depiction of the real world environment. However, without these values, further progress will not be possible. Table 1 shows the nominal values of the noise spectral densities and correlation parameter that have been selected. Three out of the five values were selected based on the reasoning of Widnall and Sinha (Ref 4). These, and the reasoning for the remaining two, are described in the following section.

Typical Values of NSD/Correlation Parameters

As suggested by Widnall and Sinha (Ref 4), a typical root mean square (rms) amplitude, for a short correlation time acceleration error, is about 200 μg. This figure is appropriate assuming a horizontal maneuver of duration of 60 seconds. This error could be caused by

(1) A 200 μrad misalignment of the input axes caused by the vertical accelerometer, and

(2) a horizontal maneuver acceleration of about one g.

Assuming a repeated random maneuver, the area of the acceleration error autocorrelation is (as derived in Ref 4)

$$Q_{a1} = 2.4 \times 10^{-4} \text{ m}^2/\text{sec}^3 \tag{39}$$

The area of the autocorrelation is the low frequency value of the power spectral density. For a white gaussian noise whose autocorrelation is the dirac delta function with area Q_{a1} , the spectral density applies at all frequencies (Ref 4). Since we are interested in the lower frequencies of the short-correlation acceleration error, therefore, the low frequency density of Eq (39) is used for the spectral density of the white noise for all frequencies.

The acceleration error δa (Fig. 4) models the inertial vertical acceleration error and it is caused by reasons already outlined earlier in this chapter. For an assumed period of 1000 seconds, if the rms value of the accelerometer bias is expected to shift 100 μg approximately, then the strength of the white noise w_{a2} as derived in Ref 4 is

$$Q_{a2} = 1.0 \times 10^{-9} \text{ m}^2/\text{sec}^5 \tag{40}$$

For a short correlation time altimeter error, it is assumed that an rms error of 10m may be present in the baro-altitude with a correlation time of one second (Ref 4). Thus the strength of the white gaussian noise w_{b1} as derived in Ref 4 is

$$Q_{b1} = 100 \text{ m}^2\text{sec} \tag{41}$$

As explained earlier in this chapter, the white noise w_{b2} models the error state δb (Fig. 4) which represents the baro-altimeter error which is the sum of many terms (Eq (5)). For the kind of trajectory (see Chapter IV) and for the minimization of the altitude error at the TERCOM-update, the altimeter bias or the standard setting error is of primary concern. It is assumed that the vehicle has been in flight for a sufficiently long time over a great distance before the TERCOM-update and the effect of the standard setting error is predominant. If the altimeter bias is represented by e_{po} (consistent with Eq (5)), then this error can be modeled as a first-order Markov process given by (Ref 4)

$$\dot{e}_{po} = -a\, e_{po} + w_{b2} \tag{42}$$

$$a = V/d_{alt} \tag{42a}$$

$$Q_{b2} = 2\, a\, \sigma_{alt}^{2} \tag{42b}$$

where d_{alt} is the correlation distance of the weather system, σ_{alt} is the standard deviation of the variation in altitude of a constant pressure surface, Q_{b2} is the power spectral density of the white gaussian noise w_{b2}, and V is the vehicle speed. For a vehicle speed of 600 miles/hr (more appropriate for a missile), correlation distance (d_{alt}) of 250 nautical miles and a one-sigma value (σ_{alt}) of 500 feet (Ref 4), the strength of the white noise and the value of the correlation parameter become

$$a = \frac{600 \times 5280}{3600 \times 250 \times 6080\ 20} \ \text{sec}^{-1} \tag{43}$$

or

$$a = 5.793 \times 10^{-4}/\text{sec} \tag{43a}$$

and

$$Q_{b2} = (2)\ (5.79 \times 10^{-4})\ (152.4\ \text{m})^2 \tag{44}$$

or

$$Q_{b2} = 26.91\ \text{m}^2\text{sec}^{-1} \tag{44a}$$

Equations (43) and (43a) are valid only for the constant velocity of 600 mi/hr, and the value of the correlation parameter (a) will change with change in velocity of the vehicle, as for example during descents or ascents.

As stated previously, Table 1 shows the values of the spectral densities and correlation parameter for use in Eq (38).

TABLE II-1

Nominal Values of Noise Spectral Densities and Correlation Parameter

White Noise for	Noise Density/ Correlation Parameter	Value
Short Correlation Time Acceleration Error	Q_{a1}	$2.4\text{x}10^{-4}\ m^2 sec^{-3}$
Acceleration Error Random Walk	Q_{a2}	$1.0\text{x}10^{-9}\ m^2 sec^{-5}$
Short Correlation Time Altimeter Error	Q_{b1}	$100\ m^2 sec$
Altimeter Error First-Order Lag	Q_{b2}	$26.91\ m^2 sec^{-1}$
Correlation Time for First-Order Lag	a	$5.793\text{x}10^{-4}\ sec^{-1}$

III. Program for Minimization of Cost

Selection and Development of Routine

The previous sections dealt with the development of the cost function as given in Eq (19). The parameters which need to be optimized are the three loop gains (K_1 , K_2 , K_3) of the vertical channel. For convenience's sake, the cost function of Eq (19) is reproduced as

$$J(K_1,K_2,K_3) = J_1(K_1,K_2,K_3) + J_2(K_1,K_2,K_3) \tag{45}$$

$$J(K_1,K_2,K_3) = \int_{t_3}^{t_4} (\delta h_1)^2 dt + (t_4 - t_3)\ (P_{\delta h}) \tag{45a}$$

where the second-half portion (i.e., the covariance $P_{\delta h}$ or $\overline{(\delta h_2)^2}$ since the mean of stochastic inputs is zero) of the right hand side is given by Eq (38) and the first one by Eqs (1), (2), and (3). To achieve the total cost, Eqs (1) through (3) need to be integrated for each interval of time to which Eq (38) is added. Thus, an integration package is required along with the search routine.

The integration and search routines selected for this thesis were DGEAR and ZXMIN, respectively, both of which reside in the IMSL Library (reference). The DGEAR routine finds approximations to the solution of a system of first order ordinary differential equations with initial conditions. The basic method used for the solution is of implicit linear multistep type. This routine is very useful in solving the stiff differential equations which were encountered during the course of this thesis (small step sizes were taken by the integration routine to achieve reasonable accuracy for extremely large values of the gain K_1). References 10 and 11 can be consulted for more details. The search routine ZXMIN is based on the Harwell library routine VA10A and utilizes the quasi-Newton method to find the minimum of a function. The search routine ZXMIN was selected because it requires no explicit gradient information from the user (it internally computes the gradient if not available). Reference 11 and the IMSL package can be consulted for additional information.

A simple flow chart of the computer program is shown in Figure 9. Estimates of the three loop gains are fed into the search routine which outputs the values of the cost and the three loop gains. The search routine iteratively estimates the values of the loop gains until a minimized cost is obtained. The convergence condition is satisfied if, on two successive iterations, the parameter estimates (i.e., K_1, K_2, K_3) agree component by component to the number of significant digits specified (3 to 5

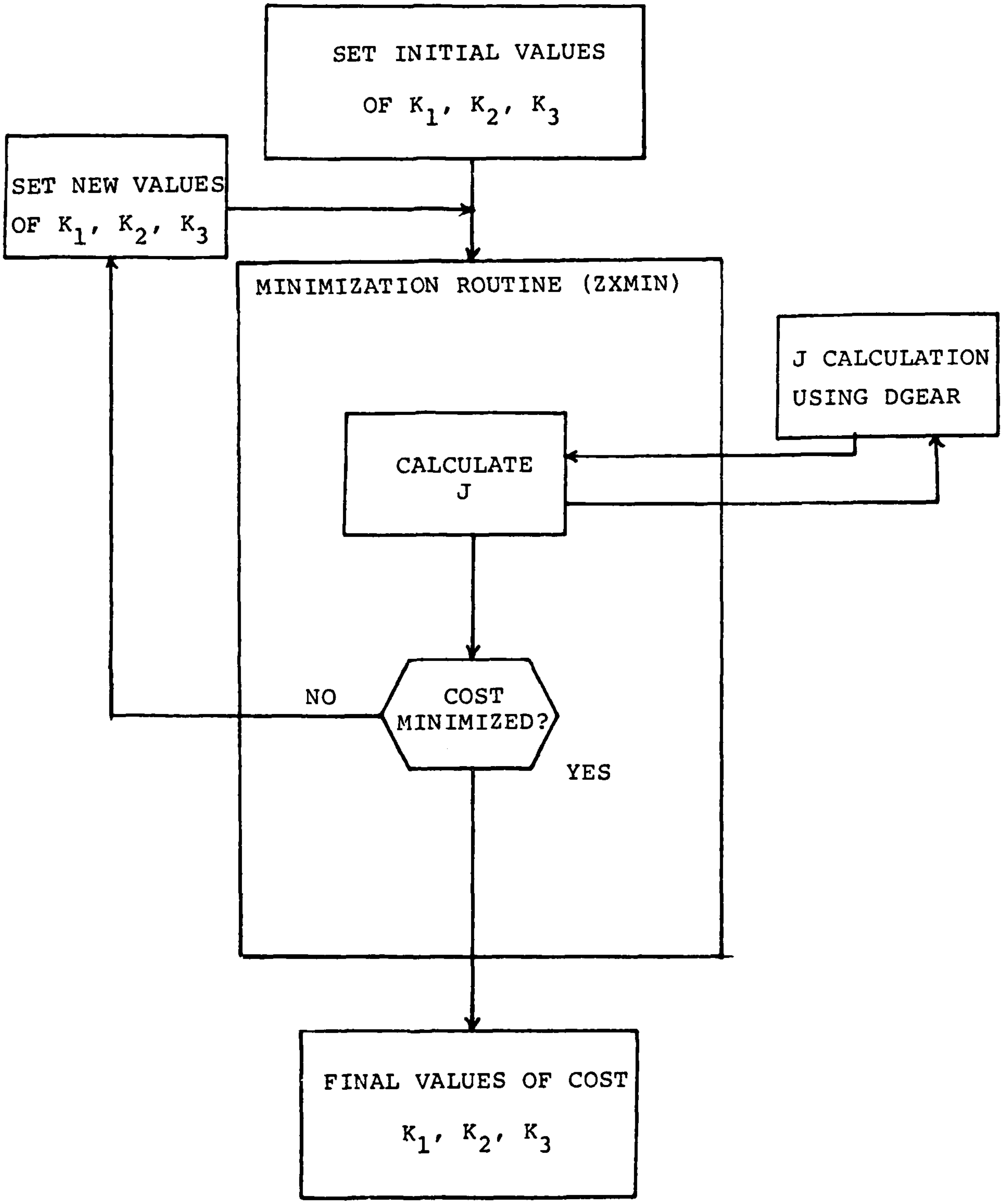


Figure 9. Flow Diagram of Search Routine

significant digits for this study). A sample listing of the program can be found in Appendix B.

Validity Check of Program

The first step after the development of the minimization program was to validate it through an earlier published result. The results obtained by Widnall and Sinha (Ref 4) were selected for this comparison.

To do this, the cost function becomes

$$J(v) = \overline{(\delta v)^2} \tag{46}$$

where $\overline{(\delta v)^2}$ is given by an equation developed along the lines similar to Eq (38) for the case of the mean squared value of the vertical velocity error as done by Widnall and Sinha (Ref 4).

Using the values for the strength of the white noise as given by Widnall and Sinha (Ref 4), the results obtained using the method developed above for the three loop gains were exactly the same as those obtained by them, thus validating the minimization program.

Scaling and Techniques Used

It is well known that one of the greatest pitfalls of a computer search routine is that the routine is liable to converge toward a local minimum, whereas what is needed is the global minimum of the function. Unless the function to be minimized is well defined (in which case the local minimum, different from a global minimum, does not exist), the results obtained from a computer search are often questionable. To overcome this problem, it is advisable to have many sets of starting points for the input variables.

For this thesis, the input variables, as stated earlier, are the three loop gains (K_1, K_2, K_3) of the vertical channel. Dr. Widnall and Sinha (Ref 4) discovered through dimensional analysis a correct expression that also gives an approximate value for the gain K_1 :

$$K_1 = \sqrt{\frac{Q_{b2}}{Q_{n1}}} \tag{47}$$

where Q_{b1} and Q_{b2} are the strength of the white noises associated with w_{b1} and w_{b2} , respectively (see Fig. 3). With the approximate value of K_1 known, it was relatively easy to implement the search routine. Nonetheless, different starting points for gains K_2 and K_3 were tried to ensure a global minimum. Since this routine also utilized an integration package, it was well worth the effort to keep a tight control on the tolerance ($\leq 10^{-8}$) in the integration.

Another problem encountered was the fact that the values of the three loop gains differ from each other by a large magnitude. Consequently, it was necessary to scale the three gains to the same level before they were fed into the search routine. This procedure ensured that the same number of significant digits was obtained in the final values of the loop gains and it also simplified the job of the optimization algorithm.

With the cost developed in Chapters I and II, and the search routine developed above, values for the optimum loop gains were found (see Chapter V). It was now necessary to validate these optimum gains through a simulated flight of a vehicle which is presented in the following chapter.

IV. Error State Propagation and Simulation

The Truth Model

A "truth model" is the analytic designer's best description of the real world behavior of the INS. In this section, a 50 state system error model (or truth model), which is needed for Monte Carlo study of optimal gains (covariance analysis program was not used due to non-zero mean disturbance since one of the requirements of covariance analysis is that all inputs should be of zero mean), is presented in the form of a stochastic linear vector differential equation as shown in Eq (48).

$$\dot{\underline{x}}(t) = \underline{F}(t)\underline{x}(t) + \underline{G}(t)\underline{w}(t) \qquad (48)$$

where

$\underline{x}(t)$ is the 50 dimensional state vector,

$\underline{F}(t)$ is the (50x50) error propagation matrix,

$\underline{w}(t)$ is a (10x1) vector of white noise forcing functions, and

$\underline{G}(t)$ is a (50x10) input matrix.

The error model of 50 state variables documented in this thesis is the Litton LN-15 navigation system (Ref 7) with the platform oriented in East-North-Up (ENU) local-level frame. The 50 state variables are presented in Table IV-1. Variables 1 to 9 are the basic position velocity and attitude variables. Variable 10 is the additional integration in the altitude channel mechanization. Variables 11 to 43 are the gyro and accelerometer sources of error. Variables 44 to 50 are the altimeter errors and gravity disturbances. Variables 11 to 50 are modeled as random constants, random walks and first-order Markov processes. The models are briefly summarized in this thesis. Details are available in Reference 8.

A random constant is modeled as the output of an integrator with zero input and an initial condition which has a zero mean (could be non-zero mean) and a variance P_o . The model is suitable for an instrument bias that changes each time the instrument is turned on, but remains constant while the instrument is on.

The random walk model is the output of an integrator driven by a zero mean white gaussian noise. The defining equations are

$$\dot{x}(t) = w(t) \qquad x(t_o) = 0 \tag{49}$$

$$E\{w(t)\} = 0 \tag{50}$$

$$E\{w(t)w(t+\tau)\} = Q(t)\delta(\tau) \tag{51}$$

TABLE IV-1

Error Model State Variables

Basic Inertial Navigation Errors		
1.	$\delta\lambda$	Error in east longitude
2.	δL	Error in north latitude
3.	δh	Error in altitude
4.	δV_e	Error in east velocity
5.	δV_n	Error in north velocity
6.	δV_z	Error in vertical velocity
7.	ε_e	East attitude error
8.	ε_n	North attitude error
9.	ε_z	Vertical attitude error
Vertical Channel Error Variable		
10.	$\delta\hat{a}$	Vertical acceleration error variable in altitude channel
G-Insensitive Gyro Drifts		
11.	DX_f	x-gyro drift rate
12.	Dy_f	y-gyro drift rate
13.	DZ_f	z-gyro drift rate

TABLE IV-1 (Cont'd)

G-Sensitive Gyro Drift		
14.	DX_x	x-gyro input axis g-sensitivity
15.	DX_y	y-gyro spin axis g-sensitivity
16.	Dy_x	y-gyro spin axis g-sensitivity
17.	Dy_y	y-gyro input axis g-sensitivity
18.	DZ_y	z-gyro spin axis g-sensitivity
19.	DZ_z	z-gyro input axis g-sensitivity
G^2-Sensitive Gyro Drift Coefficients		
20.	DX_{xy}	x-gyro spin input g^2-sensitivity
21.	DY_{xy}	y-gyro spin input g^2-sensitivity
22.	DZ_{yz}	z-gyro spin input g^2-sensitivity
Gyro Scale Factor Errors		
23.	GSF_x	x-gyro scale factor error
24.	GSF_y	y-gyro scale factor error
25.	GSF_z	z-gyro scale factor error
Gyro Input Axis Misalignments		
26.	XG_y	x-gyro input axis misalignment about y
27.	XG_z	x-gyro input axis misalignment about z
28.	YG_x	y-gyro input axis misalignment about x
29.	YG_z	y-gyro input axis misalignment about z
30.	ZG_x	z-gyro input axis misalignment about x
31.	ZG_y	z-gyro input axis misalignment about y

TABLE IV-1 (Cont'd)

Accelerometer Biases

32.	AB_x	x-accelerometer bias
33.	AB_y	y-accelerometer bias
34.	AB_z	z-accelerometer bias

Accelerometer Scale Factor Errors

35.	ASF_x	x-accelerometer scale factor error
36.	ASF_y	y-accelerometer scale factor error
37.	ASF_z	z-accelerometer scale factor error

Accelerometer Input Axis Misalignment

38.	XA_y	x-accelerometer input axis misalignment about y
39.	XA_z	x-accelerometer input axis misalignment about z
40.	YA_x	y-accelerometer input axis misalignment about x
41.	YA_z	y-accelerometer input axis misalignment about z
42.	ZA_x	z-accelerometer input axis misalignment about x
43.	ZA_y	z-accelerometer input axis misalignment about y

Barometric Altimeter Error

44.	e_{po}	Error due to variation in altitude of a constant pressure surface

TABLE IV-1 (Cont'd)

Gravity Uncertainties		
45.	δg_e	East deflection of gravity
46.	δg_n	North deflection of gravity
47.	δg_z	Gravity anomaly
***Additional Baro-Inertial Altimeter Errors**		
48.	e_{hsf}	Scale factor error
49.	C_{sp}	Coefficient of static pressure measurements
50.	τ_b	Altimeter lag

*These states were grouped separately from state 44 to conform to those given in Reference 7.

If Q(t) is constant, then

$$E\{x^2(t)\} = (t-t_o)Q \tag{52}$$

where Q is the strength of the white gaussian noise, and $\delta(\tau)$ is the delta function.

The random walk is a useful model for errors that grow without bound or may vary slowly (or unexpectedly as due to instrument failure or degradation).

A first order Markov model is the output of a first order lag driven by a zero mean white gaussian noise of strength Q . The equations are

$$\dot{x}(t) = -\frac{1}{T}x(t) + w(t) \tag{53}$$

$$E[x^2(t)] = QT/2 \tag{54}$$

where T is correlation time. The first order Markov model is a useful shaping filter, providing adequate approximation to a wide variety of empirically observed band-limited (wide or narrow band) noises.

The error propagation matrix of the vector differential equation governing the 50 state variables is presented in partitions in Figures 10, 11, 12, and 13. Figure 10 presents the upper (9x9) fundamental matrix (Pinson error model) of INS general error differential equations. Figure 11 shows those elements that must be added to the elements of the general 9x9 error propagation matrix of Figure 10 to obtain the partition of position, velocity, attitude and vertical-acceleration error state variables. Figure 12 presents the non-zero elements of the gyro-error columns of the error propagation matrix. Figure 13 presents the non-zero elements of the accelerometer, gravity disturbance and altimeter columns of the fundamental matrix. Notation used in the above mentioned figures is defined in Table IV-2. Error

$$\begin{array}{ll|ccc|ccc|ccc}
 & & \delta\lambda & \delta L & \delta h & \delta V_e & \delta V_n & \delta V_z & \varepsilon_e & \varepsilon_n & \varepsilon_z \\
 & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
\delta\dot{\lambda} & 1 & 0 & \rho_z/\cos L & -\rho_n/R\cos L & 1/R\cos L & 0 & 0 & 0 & 0 & 0 \\
\delta\dot{L} & 2 & 0 & 0 & \rho_e/R & 0 & 1/R & 0 & 0 & 0 & 0 \\
\delta\dot{h} & 3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\hline
\delta\dot{V}_e & 4 & 0 & F_{42} & F_{43} & F_{44} & (\omega_z+\Omega_z) & -(\omega_n+\Omega_n) & 0 & -f_z & f_n \\
\delta\dot{V}_n & 5 & 0 & F_{52} & F_{53} & -2\omega_z & -K_z & \rho_e & f_z & 0 & -f_e \\
\delta\dot{V}_z & 6 & 0 & -2\Omega_z V_e & F_{63} & 2\omega_n & -2\rho_e & 0 & -f_n & f_e & 0 \\
\hline
\dot{\varepsilon}_e & 7 & 0 & 0 & -\rho_e/R & 0 & -1/R & 0 & 0 & \omega_z & -\omega_n \\
\dot{\varepsilon}_n & 8 & 0 & -\Omega_z & -\rho_n/R & 1/R & 0 & 0 & -\omega_z & 0 & \omega_e \\
\dot{\varepsilon}_z & 9 & 0 & F_{92} & -\rho_z/R & \tan L/R & 0 & 0 & \omega_n & -\omega_e & 0 \\
\hline
\end{array}$$

Figure 10. Upper 9x9 Partition of $\underline{F}(t)$

	$\delta\lambda$ 1	δL 2	δh 3	δV_e 4	δV_n 5	δV_z 6	ε_e 7	ε_n 8	ε_z 9	$\delta\hat{a}$ 10	e_{nsf} 48
$\delta\dot{\lambda}$ 1											0
$\delta\dot{L}$ 2					0			0		0	0
$\delta\dot{h}$ 3			$-K_1$								K_1h
$\delta\dot{V}_e$ 4											0
$\delta\dot{V}_n$ 5					0			0			0
$\delta\dot{V}_z$ 6			$-K_2$							-1	K_2h
$\dot{\varepsilon}_e$ 7											
$\dot{\varepsilon}_n$ 8		0			0			0		0	0
$\dot{\varepsilon}_z$ 9											
$\delta\dot{\hat{a}}$ 10			K_3		0			0		0	$-K_3h$
$\dot{e}_{hsf}$ 48		0			0			0		0	0

Figure 11. Added Elements to Figure 10 of LN-15 Fundamental Matrix

		DX_f	DY_f	DZ_f	DX_x	DX_y	DY_y	DY_y	DZ_y	DZ_z	DX_{xy}	DY_{xy}	DZ_{yz}	GSF_x	GSF_y	GSF_z	XG_y	XG_z	YG_x	YG_z	ZG_z	ZG_y
		11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$\delta\dot{\lambda}$																						
$\delta\dot{L}$																						
$\delta\dot{h}$			0				0					0			0				0			
$\delta\dot{V}_e$																						
$\delta\dot{V}_n$			0				0					0			0				0			
$\delta\dot{V}_z$																						
$\dot{\varepsilon}_e$	7	F_{711}	F_{712}	0	F_{714}	F_{715}	F_{716}	F_{717}	0	0	F_{720}	F_{721}	0	F_{723}	F_{724}	0	F_{726}	F_{727}	F_{728}	F_{729}	0	0
$\dot{\varepsilon}_n$	8	F_{811}	F_{812}	0	F_{814}	F_{815}	F_{816}	F_{817}	0	0	F_{820}	F_{821}	0	F_{823}	F_{824}	0	F_{826}	F_{827}	F_{828}	F_{829}	0	0
$\dot{\varepsilon}_z$	9	0	0	1	0	0	0	0	f_y	f_z	0	0	F_{922}	0	0	Ω_z	0	0	0	0	ω_y	$-\omega_x$
$\delta\dot{\hat{a}}$	10		0				0					0			0				0			

Figure 12. Partition of Gyro-Error State Variables

		AB_x 32	AB_y 33	AB_z 34	ASF_x 35	ASF_y 36	ASF_z 37	XA_y 38	XA_z 39	YA_x 40	YA_z 41	ZA_x 42	ZA_y 43	e_{PO} 44	δg_e 45	δg_n 46	δg_z 47	e_{nsf} 48	C_{SP} 49	τ_b 50
$\delta\dot{\lambda}$	1													0				0	0	0
$\delta\dot{L}$	2		0			0								0		0		0	0	0
$\delta\dot{h}$	3													K_1				K_1h	F_{349}	F_{350}
$\delta\dot{V}_e$	4	F_{432}	F_{433}	0	F_{435}	F_{436}	0	F_{438}	F_{439}	F_{440}	F_{441}	0	0	0	1	0	0	0	0	0
$\delta\dot{V}_n$	5	F_{532}	F_{533}	0	F_{535}	F_{536}	0	F_{538}	F_{539}	F_{540}	F_{541}	0	0	0	0	1	0	0	0	0
$\delta\dot{V}_z$	6	0	0	1	0	0	f_z	0	0	0	0	$-f_y$	f_x	K_2	0	0	1	K_2h	F_{649}	F_{650}
$\dot{\varepsilon}_e$	7																			
$\dot{\varepsilon}_n$	8													0		0		0	0	
$\dot{\varepsilon}_z$	9																			
$\delta\hat{a}$	10													$-K_3$		0		$-K_3h$	F_{1049}	F_{1050}
e_{PO}	44													F_{4444}		0		0	0	
δg_e	45														F_{4545}	0	0			
δg_n	46													0	0	F_{4646}	0	0	0	
δg_z	47														0	0	F_{4747}			

Figure 13. Partitions of F-Matrix of Accelerometer, Altimeter and Gravity Disturbance State Variables

TABLE IV-2

Notation Used in Figure 10 to Figure 13

L	Latitude of Vehicle
Ω	Earth rotation rate
R	Radius of Earth
g	Gravity vector magnitude
V_e, V_n, V_z	Vehicle velocity with respect to earth
f_e, f_n, f_z	Components of specific force
$\Omega_n = \Omega\cos L$ $\Omega_z = \Omega\sin L$	Components of earth rate
$\rho_e = V_n/R$ $\rho_n = V_e/R$ $\rho_z = V_e \frac{\tan L}{R}$	Components of angular velocity of ENU frame with respect to earth
$\omega_e = \rho_e$ $\omega_n = \rho_n+\Omega_n$ $\omega_z = \rho_z+\Omega_z$	Components of angular velocity of ENU frame with respect to inertial space
$K_z = V_z/R$	

$$F_{42} = 2(\Omega_n V_n+\Omega_z V_z) + \rho_n V_n/\cos^2 L$$

$$F_{43} = \rho_z\rho_e + \rho_n K_z$$

$$F_{44} = -\rho_e \tan L - K_z$$

$$F_{52} = -2\Omega_n V_e - \rho_n V_e/\cos^2 L$$

TABLE IV-2 (Cont'd)

$F_{53} = \rho_n\rho_z - \rho_e K_z$

$F_{63} - 2g/R - (\rho_n^2 - \rho_e^2)$

$F_{92} = \omega_n + \rho_z \tan L$

K_1, K_2, K_3 Vertical channel loop gains

h Height of the vehicle over earth

$F_{711} = \cos\alpha$, cosine of wander angle

$F_{712} = \sin\alpha$, sine of wander angle

$F_{811} = \sin\alpha$, sine of wander angle

$F_{812} = \cos\alpha$, cosine of wander angle

f_x, f_y, f_z Components of specific force

$F_{714} = f_x \cos\alpha$

$F_{814} = f_x \sin\alpha$

$F_{715} = f_y \cos\alpha$

$F_{815} = f_y \sin\alpha$

$F_{716} = -f_x \sin\alpha$

$F_{816} = f_x \cos\alpha$

$F_{717} = -f_y \sin\alpha$

$F_{817} = f_y \cos\alpha$

$F_{720} = f_x f_y \cos\alpha$

$F_{820} = f_x f_y \sin\alpha$

TABLE IV-2 (Cont'd)

F_{721} = $-f_x f_y \sin \alpha$

F_{821} = $f_x f_y \cos \alpha$

F_{922} = $f_y f_z$

ω_x, ω_y = Components of angular velocity of ENU frame with respect to inertial space along LN-15 x,y axes

Ω_z = Up component of earth rate

d_{alt} = Correlation distance of altimeter error

d_{ge}, d_{gn}, d_{gz} = Correlation distances of gravity deflections and anomaly

σ_{alt} = 1σ amplitude of altimeter error e_{po}

$\sigma_{ge}, \sigma_{gn}, \sigma_{gz}$ = 1σ amplitude of gravity disturbances

F_{723} = $\omega_x \cos \alpha$

F_{823} = $\omega_x \sin \alpha$

F_{724} = $-\omega_y \sin \alpha$

F_{824} = $\omega_y \cos \alpha$

F_{726} = $\Omega_z \cos \alpha$

F_{826} = $\Omega_z \sin \alpha$

F_{727} = $-\omega_y \cos \alpha$

F_{827} = $-\omega_y \sin \alpha$

F_{728} = $\Omega_z \sin \alpha$

F_{828} = $-\Omega_z \cos \alpha$

TABLE IV-2 (Cont'd)

$F_{729} = -\omega_x \sin \alpha$

$F_{829} = \omega_x \cos \alpha$

v = ground speed of vehicle

$F_{349} = K_1 V^2$

$F_{350} = -K_1 V_z$

$F_{432} = \cos \alpha$

$F_{532} = \sin \alpha$

$F_{433} = -\sin \alpha$

$F_{533} = \cos \alpha$

$F_{435} = f_x \cos \alpha$

$F_{535} = f_x \sin \alpha$

$F_{436} = -f_y \sin \alpha$

$F_{536} = f_y \cos \alpha$

$F_{438} = -f_z \cos \alpha$

$F_{538} = -f_z \sin \alpha$

$F_{439} = f_y \cos \alpha$

$F_{539} = f_y \sin \alpha$

$F_{440} = -f_z \sin \alpha$

$F_{540} = f_z \cos \alpha$

$F_{441} = f_x \sin \alpha$

TABLE IV-2 (Cont'd)

$F_{541} = -f_x \cos \alpha$

$F_{649} = K_2 V^2$

$F_{650} = -K_2 V_z$

$F_{1049} = -K_3 V^2$

$F_{1050} = K_3 V_z$

$F_{4444} = -V/d_{alt}$

$F_{4545} = -V/d_{ge}$

$F_{4646} = -V/d_{gn}$

$F_{4747} = -V/d_{gz}$

TABLE IV-3

Error Source Initial Values and Statistics

Random Walks $\dot{x} = W$

State Variable	Error Source	Initial Value	Noise Spectral Density
11, 12	x and y gyro drifts	$0.5\text{x}10^{-3}$ °/hr	(0°/hr)^2/hr
13	z gyro drift	$0.7\text{x}10^{-3}$ °/hr	(0°/hr)^2/hr
32, 33, 34	x, y and z accelerometer biases	25 μg	(0 μg)^2/hr

First Order Markov Processes

$$\dot{x} = -\beta x + W \quad ; \quad N_W = 2\beta\sigma^2$$

State Variable	Error Source	Initial Value	Noise Spectral Density	Correlatio Parameter
44	Baro Altimeter Bias	50 ft	500 ft^2/nm	250 nm
45	East Gravity Deflection	26 μg	140 μg^2/nm	10 nm
46	North Gravity Deflection	17 μg	58 μg^2/nm	10 nm
47	Gravity Anomaly	35 μg	41 μg^2/nm	60 nm

(Continued on next page)

TABLE IV-3 (Cont'd)

Random Constants $\dot{x} = 0$		
State Variable	Error Source	Initial Value
14 to 19	G-sensitive gyro drift coefficients	0.5×10^{-3} °/hr/g
20 to 22	G^2-sensitive gyro drift coefficients	$0°/hr/g^2$
23 to 25	x, y and z gyro scale factor	5 ppm
26 to 31	Gyro input axis misalignment	2.5 arc sec
35 to 37	Accelerometer scale factor errors	25 ppm
38 to 43	Accelerometer input axis misalignments	5.51 arc sec
48	Altimeter scale factor	0.003
49	Static Pressure Measurement Error	$\frac{0.1540 \times 10^{-3} \text{ ft}}{(\text{ft/sec})^2}$
50	Altimeter lag	0.25 sec

source initial values and statistics are summarized in Table IV-3.

Not included in the above is the effect of the disturbance to the baro-inertial vertical channel. For this, it is necessary to show how each error associated with the vertical channel is modeled.

Recalling from Chapter I, the closed loop altitude error δh, the vertical velocity error δv_z and vertical acceleration $\delta\hat{a}$ estimate are

$$\delta\dot{h} = \delta v_z - K_1(\delta h - \delta h_b) \tag{55}$$

$$\delta\dot{v}_z = \left(\frac{2g}{R}\right)\delta h - K_2(\delta h - \delta h_b) - \delta\hat{a} \tag{56}$$

$$\delta\hat{a} = K_3(\delta h - \delta h_b) \tag{57}$$

where δh_b is the baro-altimeter error and is given by

$$\delta h_b = e_{po} + he_{hsf} + c_{sp}v^2 - \tau_b v_z + \delta D \tag{58}$$

where

h = vehicle altitude

v = vehicle speed

v_z = vertical velocity

e_{po} = altimeter bias

e_{hsf} = altimeter scale factor

c_{sp} = static pressure coefficient

τ_b = barometric time delay

δD = disturbance input to vertical channel.

The altimeter bias e_{po}, more commonly known as standard setting error or variation in height of a constant pressure surface, varies slowly due to two reasons.

(1) motion of the vehicle through the weather pattern;

(2) motion of the weather system.

The rms variation of this altitude has a bounded magnitude. In this thesis, this error is modeled as a first order lag given by

$$\dot{e}_{po} = -\omega_{alt} \, e_{po} + w_{po} \qquad (59)$$

$$\omega_{alt} = v/d_{alt}$$

$$N_{alt} = 2\omega_{alt} \, \sigma^2_{alt} \qquad (61)$$

where

d_{alt} = correlation distance of the weather

σ_{alt} = one-sigma value of the variation of altitude of a constant pressure surface

v = vehicle speed

N_{alt} = power spectral density of the white gaussian noise w_{po}

The altimeter scale factor e_{hsf} is the error due to deviation of the atmospheric temperature from the assumed temperature profile (Ref 7). The indicated altitude error (e_{temp}) is of the form

$$e_{temp} = (e_{hsf})(h) \tag{62}$$

Thus it can be viewed as an altimeter scale factor error. This error varies slowly with location and time, and it is assumed to be constant over a typical navigation flight duration. Thus it is modeled as a random constant

$$\dot{e}_{hsf} = 0 \tag{63}$$

with appropriate standard deviation.

The altitude indicated by the barometer is based on the static pressure. The latter is taken from the pressure measurements made by the pitot-static tube in the vehicle. The altimeter error e_{sp} due to the erroneous interference of the static pressure is (Ref 5)

$$e_{sp} = c_{sp}\, v^2 \tag{64}$$

The coefficient c_{sp} is nearly constant with altitude and thus it is modeled again as a random constant

$$\dot{c}_{sp} = 0 \tag{65}$$

with appropriate standard deviation.

The barometric time delay τ_b represents the time required by the static pressure in the cavity of the pressure transducer to adjust to the static pressure at port by flow of air through tubing during vehicle maneuvers (Ref 6). This time constant is nearly invariant and is modeled as

$$\dot{\tau}_b = 0 \tag{66}$$

with a given standard deviation.

The disturbance input δD is the cumulative effect of all other sources of error which may influence the vertical channel during the vehicle climbs and descents. This process has not been modeled as an additional error state; instead, it is treated deterministically. That is, it is fed to the vertical channel during the time when the vehicle performs a descent. The magnitude of this error was selected as 200 meters based on discussions with the sponsor of this thesis. Thus

$$\delta D = 200 \text{ m} \qquad t_2 \le t \le t_3 \tag{67}$$

where $(t_3 - t_2)$ is the time during which the vehicle descends.

Trajectory Selection

The trajectory generator is needed to give position, velocity and specific force throughout the interval of study. General trajectory generation programs are available; however, in this thesis it was decided to conserve the computer resources, and a trajectory of a great circle path was generated by a set of closed-form expressions for position, velocity and specific force (Ref 12).

The pattern of the flight path selected was a straight and level flight at 600 miles per hour at 11,000 feet for a duration of 500 seconds, followed by a dive at the rate of 6000 feet per minute for 100 seconds (with a corresponding decrease in ground speed), and finally leveling at 1000 feet. The altitude profile generated is shown in Figure 14, with the vertical velocity as shown in Figure 15.

The mission scenario envisions a disturbance input to the vertical channel at t = 500 seconds just at the time the vehicle starts descending, and ends at t = 600 seconds as the vehicle levels off. At time t = 610 seconds, the vehicle is required to perform a TERCOM-update ending at time t = 660 seconds. It is during this last section of

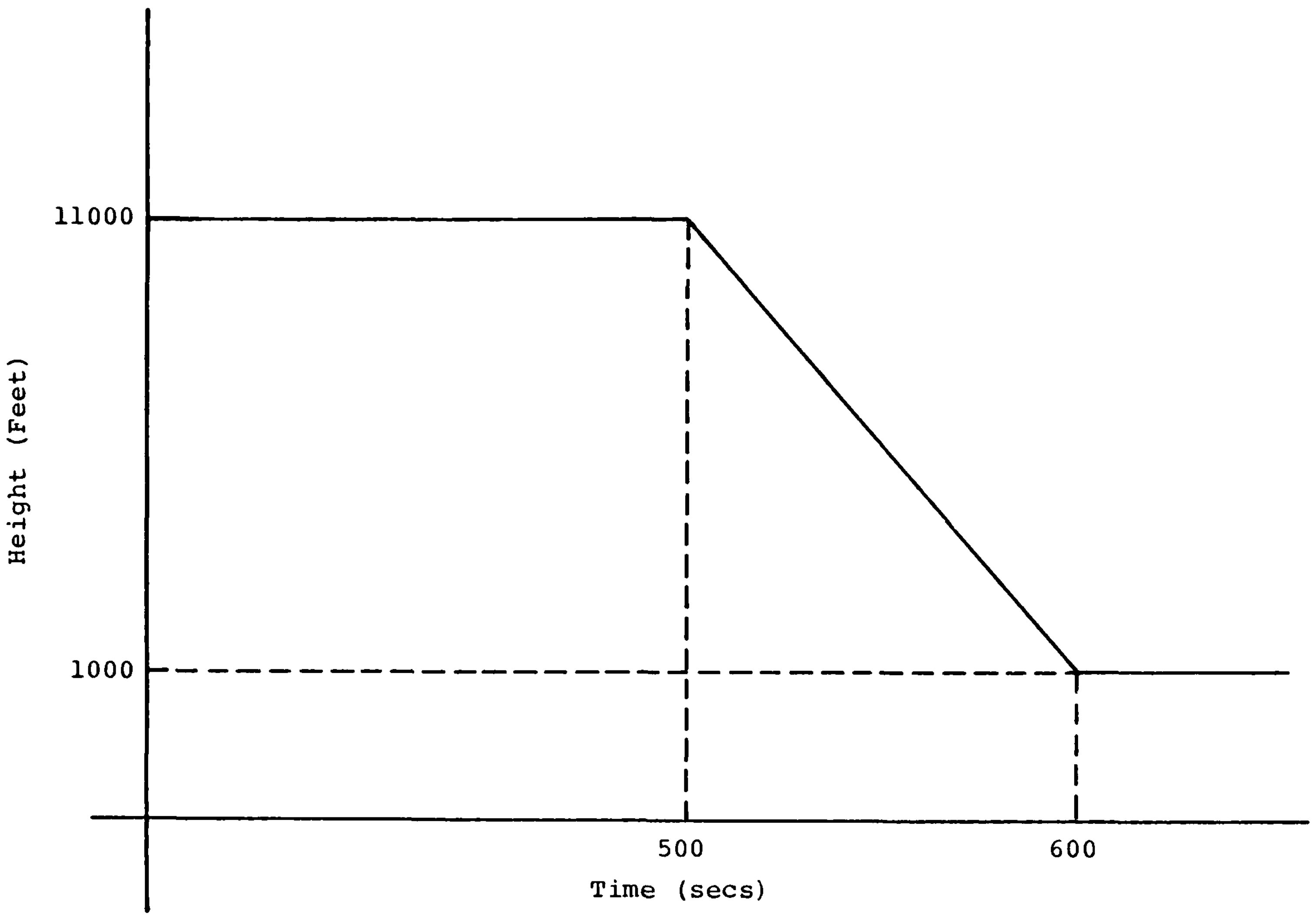


Figure 14. Altitude Profile

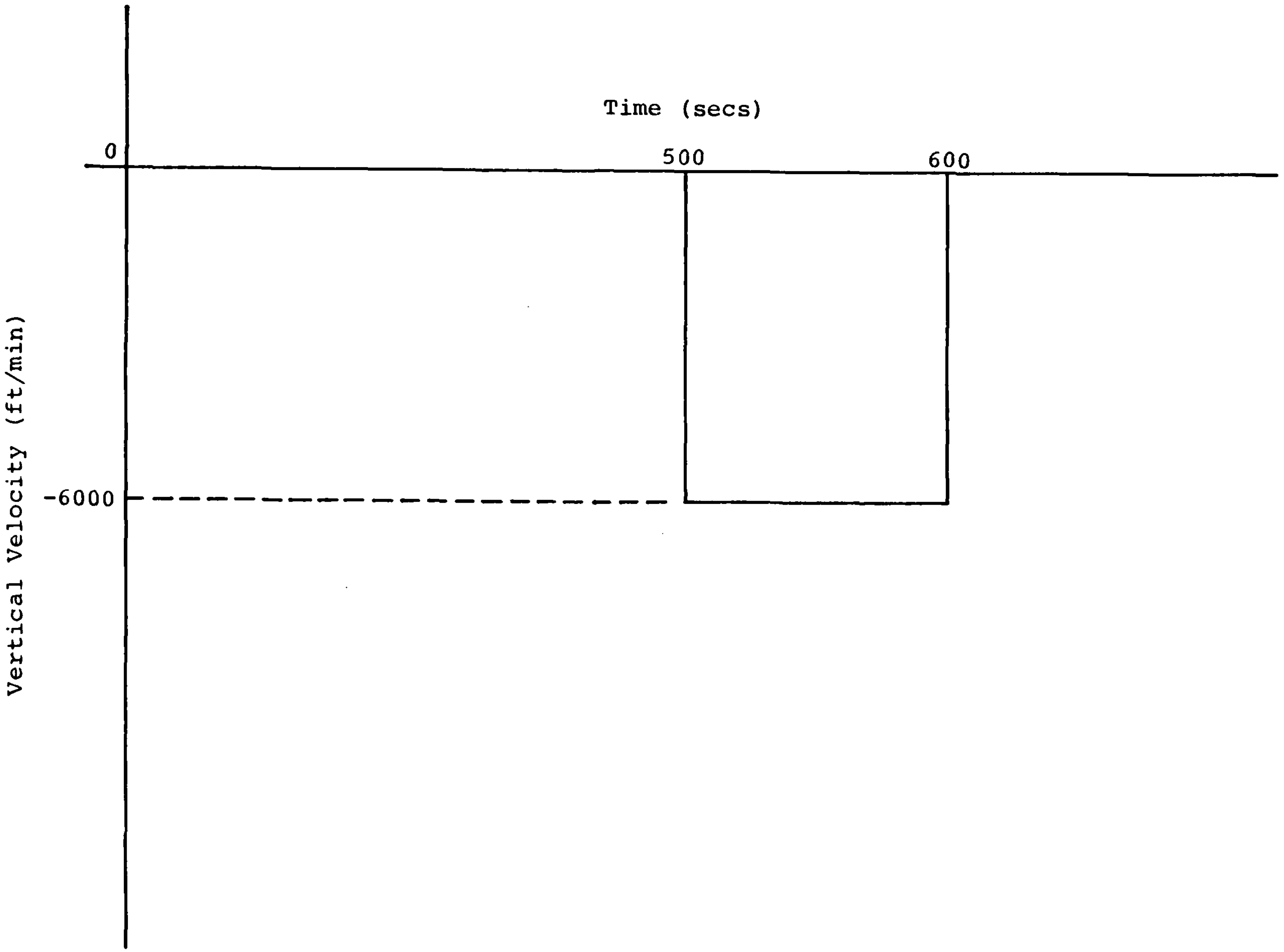


Figure 15. Vertical Velocity Profile

that the behavior of the vertical channel is analyzed. The various time intervals were given by the thesis sponsor.

Monte Carlo Simulation

A generalized Monte Carlo simulation program (SOFE - a generalized digital simulation for optimal filter evaluation) (Ref 13) was used to propagate the error states over the total time interval of 700 seconds (covariance analysis program was not done due to non-zero mean disturbance input). The companion plot program, SOFEPL, was used for generating subsequent plots (Ref 14).

SOFE requires both the true error states and the filter error states. Since there were no filter error states for this work, a dummy filter error state was programmed. The various user input routines to SOFE for one set of vertical channel gains are given in Appendix C.

The integrator in the basic SOFE cannot handle step changes. Since the very nature of this thesis involved step changes due to disturbance, altitude and vertical velocity, these were approximated as cosine functions over a small interval of time ($\Delta t = 0.01$ secs) at each corner. This device enabled the integrator to work properly and made no substantial impact on the results.

Three plots were generated for each flight. Error states 3 (altitude error), 6 (vertical velocity error) and 44 (barometric error) (see Table IV-1) were plotted using

SOFEPL. The three plots generated were for each set of vertical loop gains; i.e., classical, improved and combined gains. Thirty Monte Carlo runs were carried out for each case to get an ensemble average. For one set of loop gains, the statistical results of 100 Monte Carlo runs were not significantly different from the results of thirty Monte Carlo runs. To conserve computer resources, thirty Monte Carlo runs were used in all the following analyses. For further insight, one Monte Carlo run was also carried out for each set of gains. For comparison purposes, the starting point of the random number generator was set to a fixed value for all three sets of gains. That is, each separate set was generated with the same noise realization. The results obtained from these simulations are given in the following chapter.

V. Results

Basic Cost Function

Recalling from Chapter II, the basic cost function was

$$J(\underline{K}) = \int_{t_3}^{t_4} (\delta h_1)^2 \, dt \qquad (68)$$

For the above cost function, the barometric data was assumed perfect with no uncertainties whatsoever (Figure 1), except during the interval $(t_2 - t_1)$ (see Figure 2) at which time the disturbance was fed into the vertical channel. Equation (68) was minimized through the search routine, and the optimum values of the three vertical loop gains are given in Table V-1.

TABLE V-1

Optimized Gains (Basic Cost Function)

Basic Cost Function $J(\underline{K}) = \int_{t_3}^{t_4} (\delta h_1)^2 \, dt$		
Vertical Loop Gains	Optimized Result	Units
K_1	935678.67	sec^{-1}
K_2	0.02	sec^{-2}
K_3	0.0001	sec^{-3}

The message from Table V-1 is apparent. With the cost function of Eq (68), the vertical loop gain K_1 needs to be set to essentially infinity regardless of the values of K_2 and K_3 . If we analyze the set-up of the cost function in more detail, the result is not surprising. As said earlier, the barometric data is assumed perfect except during the disturbance interval; i.e., the barometric data is perfect before and after the disturbance. The optimum INS altitude estimate after the disturbance would be the perfect baro output, and tight tracking control will minimize the INS variation from this ideal baro indication. So, the large value of gain K_1 can be anticipated for this highly-simplified model. If baro altitude were not perfect, then the gain K_1 would settle for a far lesser value, thereby indicating that the inertial system does not truly believe in the data from the altimeter due to uncertainties in the latter. For this case, however, the data from the altimeter is perfect, thus the gain K_1 must be set to infinity to track the altimeter without any lag. A constraint optimization routine or addition of a term in the cost function to penalize huge values of K_1 (or K_2 or K_3) could be effectively used at this stage. Such a procedure is inconsistent with the objectives of this analysis and was not pursued.

The optimized values of the three gains of Table V-1 are not very accurate because of the inherent limitations of the integration routine used; however, these values give insight into the behavior of the vertical channel.

New Cost Function

With the addition of uncertainties in the vertical channel and barometric data (Fig. 4), the cost function was

$$J(\underline{K}) = \int_{t_3}^{t_4} (\delta h_1)^2 \, dt + (t_4 - t_3) \, [\overline{(\delta h_2)^2}] \tag{69}$$

It may be of importance to note that, in the minimization of Eq (69), a weighting factor, β, can be introduced such that

$$J(\underline{K}) = (\beta) \int_{t_3}^{t_4} (\delta h_1)^2 dt + (1-\beta)(t_4-t_3)[\overline{(\delta h_2)^2}] \cdot \tag{70}$$

For a value of β (between 0 and 1), it is possible to have any combination of the mean squared error due to the disturbance and noise. This in effect scales the size of the deterministic disturbance. If β is set to 0.5, then Eq (70) reverts back to the form equivalent to Eq (69).

To provide a baseline design and performance against which to compare the optimized performance, the classical set of gains is

$$
\begin{aligned}
K_1 &= 3.0 \times 10^{-2} \quad \text{sec}^{-1} \\
K_2 &= 3.0307 \times 10^{-4} \ \text{sec}^{-2} \\
K_3 &= 1.0 \times 10^{-6} \quad \text{sec}^{-3}
\end{aligned}
\tag{71}
$$

The only rationale given for these gains is that the third-order control system has a triple pole with a 100 second time constant. This design specification allows the INS estimate to average out the high frequency barometric noise, but is not optimal in any sense. It may also be interesting to compare the performance with the optimized set of gains obtained by Widnall and Sinha (Ref 4) for the mean squared velocity error; they are

$$
\begin{aligned}
K_1 &= 1.004 \qquad \text{sec}^{-1} \\
K_2 &= 4.17 \times 10^{-3} \ \text{sec}^{-2} \\
K_3 &= 4.39 \times 10^{-6} \ \text{sec}^{-3}
\end{aligned}
\tag{72}
$$

As stated in earlier chapters, the magnitude of the disturbance input to the vertical channel was assumed to be 200 meters. With such a disturbance, the minimization of Eq (69) led to the optimized set of gains as given in Table V-2.

TABLE V-2

Optimized Gains (New Cost Function)

New Cost Function $J(h) = \int_{t_3}^{t_4} (\delta h_1)^2 dt + (t_4 - t_3)[\overline{(\delta h_2)^2}]$		
Vertical Loop Gains	Value	Units
K_1	0.631	sec^{-1}
K_2	4.78×10^{-3}	sec^{-2}
K_3	6.335×10^{-5}	sec^{-3}

The mean squared altitude error, with the values of the noise spectral densities as given in Chapter II, for the classical set of gains (Eq (71)) was found to be

$$\overline{(\delta h)^2_{CL}} = 595.879 \text{ m}^2$$

$$= (24.41)^2 \text{ m}^2 \qquad (73)$$

The corresponding mean squared altitude error for the improved set of gains of Table V-2 is

$$\overline{(\delta h)^2_{IMP}} = 54.899 \text{ m}^2$$

$$= (7.409)^2 \text{ m}^2 \qquad (74)$$

In calculating the mean squared error as given in Eqs (73) and (74), the mean square value of the error state 44 (error due to variation in altitude of a constant pressure surface) (first order lag shaping filter, Fig. 4) was subtracted so that true performance comparison could be made. between the classical and improved gains. The mean squared error for gains of Eq (72) was not calculated for reasons presented later. Accordingly, $(152.4 \text{ m})^2$ was subtracted (see Chapter II) and is not included in Eqs (73) and (74).

This performance improvement is significant relative to the classical gains. The mean squared altitude error is 70% lower.

The poles of the closed-loop portion of the vertical channel are the three roots of the characteristic equation (see Chapter II)

$$S^3 + K_1S^2 + (K_2 - 2g/R)S + K_3 = 0$$

With the values of the loop gains of Table V-2, the three poles are located at

$$P_1 = -0.6235 \text{ sec}^{-1}$$

$$P_2, P_3 = -3.75 \times 10^{-3} \pm j\, 9.36 \times 10^{-3} \text{ sec}^{-1} \tag{75}$$

These poles have a time constant of

$$\tau_1 = 1.604 \text{ sec}$$

$$\tau_2, \tau_3 = 266.67 \text{ sec} \tag{76}$$

Comparing with the classical gains, one time constant is a factor of 100 faster; the other two time constants are a factor of three slower.

The individual contributions of the various white noises (see Fig. 4) to the mean squared altitude error, for the noise densities as given in Chapter II, are shown in Table V-3.

Table V-3 shows that for the classical gains, the mean squared altitude error is dominated by the short correlation time acceleration error and more so by the altimeter error (first order lag). For the improved gains, the contribution of altimeter error and short correlation time altimeter error is the greatest.

TABLE V-3

Contribution of White Noises
to Mean Squared Altitude Error

Noise Density	Mean Squared Altitude Error $(m)^2$	
	Classical Gains	Improved Gains
Q_{a1}	15.0	0.04064
Q_{a2}	1.875	1.686×10^{-3}
Q_{b1}	2.074	31.937
Q_{b2}	576.93	22.82
TOTAL	$595.879 = (24.41)^2$	$54.899 = (7.409)^2$

Unfortunately, under the presence of the disturbance, comparison cannot be made between the optimized gains for the mean squared velocity error found by Widnall and Sinha (Ref 4) as given in Eq (72), and the improved gains for the mean squared altitude error of Table V-2. In addition, the gains of Eq (72) are the optimized gains for the vertical velocity error, whereas the improved gains of Table V-2 are for the altitude error and a performance comparison between these sets of gains would be pointless. To gain insight into the nature of the optimized gains, the disturbance was set to zero, and using the values of the dynamic driving noises used by Widnall and Sinha (Ref 4) (with the first order lag for noise w_{b2} instead of random walk as done in Ref 4), the optimized gains obtained are as given in Table V-4. The gains of Table V-4 show a considerable departure from Table V-2, especially the gains K_1 and K_3. In essence, as pointed out by Widnall and Sinha (Ref 4), the gain K_1 primarily depends on the strength of the noise sources Q_{b2} and Q_{b1}. The value of gain K_1 for Case II is in excellent agreement with the formula (Ref 4)

$$K_1 = \sqrt{\frac{Q_{b2}}{Q_{b1}}} = \sqrt{\frac{26.91}{100}} \text{ sec}^{-1} = 0.51 \text{ sec}^{-1} \qquad (77)$$

The gains of Table V-4 (for zero disturbance) are not explicitly required for this thesis since the very objective of the thesis was to find optimum gains due to non-stochastic (disturbance) and stochastic inputs. To obtain further insight into the nature of the optimized set of Table V-2, it was necessary to find their sensitivity to the time intervals t_1, t_2, t_3, and t_4 (see Fig. 2), which is analyzed in the following section.

Sensitivity Analysis

It may be recalled from Figure 2 and Chapter IV, the vehicle was required to descent for a period of 100 seconds in the interval $(t_2 - t_1)$, and perform a TERCOM-update

TABLE V-4

Optimized Gains for Zero Disturbance

Optimized Gains for Disturbance = 0	
Loop Gains	Optimized Gains
K_1	$0.54\ \text{sec}^{-1}$
K_2	$7.77 \times 10^{-3}\ \text{sec}^{-2}$
K_3	$1.02 \times 10^{-4}\ \text{sec}^{-3}$

for 50 seconds during the interval $(t_4 - t_3)$. Table V-5 shows the values of the vertical loop gains for an increase of 10% in each of the time intervals $(t_2 - t_1)$, $(t_3 - t_2)$ and $(t_4 - t_3)$, respectively. Comparing with the improved gains (Table V-2), we find that the gains are very sensitive to the time interval $(t_3 - t_2)$; i.e., after the descent and before the TERCOM-update. On examining it more closely, we find that, increasing the time interval after the disturbance interval $(t_2 - t_1)$, it is logical for these gains to change and settle on the steady-state values, because the effect of the disturbance is decreasing; in effect, had not the gains been optimized for the interval $(t_4 - t_3)$, they would have approached the values as given in Table V-4.

TABLE V-5

Sensitivity of Gains

Sensitivity Analysis			
Loop Gains	Interval (t_2-t_1) increased 10%	Interval (t_3-t_2) increased 10%	Interval (t_4-t_3) increased 10%
K_1	0.632 sec^{-1}	0.557 sec^{-1}	0.65 sec^{-1}
K_2	4.63×10^{-3} sec^{-2}	3.32×10^{-3} sec^{-2}	4.9×10^{-3} sec^{-2}
K_3	6.47×10^{-5} sec^{-3}	5.81×10^{-5} sec^{-3}	5.21×10^{-5} sec^{-3}

It is natural for the gains to approach their steady-state value in the long run, once the effect of the disturbance is over. In addition, the very slight difference between the intervals $(t_2 - t_1)$, $(t_4 - t_3)$ and those of Table V-2 is due to the minimal effect of the disturbance.

The optimized gains for the disturbance (Table V-2) were checked out in the simulated flight of a vehicle performing a TERCOM-update and the results are shown in the next section.

Simulation Results

As stated in Chapter IV, one flight profile was used with three different gain sets; classical, improved and combined. In the first flight, only the classical gains were available throughout the duration of 700 seconds, and in the second flight, the improved gains of

Table V-2 were programmed. For the third flight, the vertical channel was programmed to use the classical gains up to the time t_3 = 610 seconds, and then switched over to the improved gains of Table V-2 for the TERCOM-update interval of $(t_4 - t_3)$ = 50 seconds, and finally switched back to the classical gains after the time t_4 (see Fig. 2). The gains of Table V-2 were optimized only for the duration of the TERCOM-update; therefore, it was appropriate to program them only for this interval. The results for the altitude error for the classical, improved and combined flights are shown in Figures 16, 17 and 18.

On examining Figure 16 (classical gains), we see an initial hump around time t = 100 seconds with the error finally settling to its steady state value at time t = 400 seconds. This slow rise to its transient peak around t = 100 seconds is due to the inherent lag (τ = 100 secs) in the classical gains. Thus, with these gains it takes a long time to build up the error and settle on the steady state value. At time t = 500 seconds, the vehicle went into a dive, and the buildup of the error after t = 500 seconds due to error in altimeter is evident. At time t = 600 seconds, the descent of the vehicle stops, but the error in altitude takes a long time to descend down and follow the altimeter. Notice at time t = 610 seconds, where the vehicle was required to perform the TERCOM-update, the altitude error is still in excess of 800 feet. This error may seem unrealistic, but for the kind of disturbance and the flight trajectory the result is not incorrect. For a disturbance input of a smaller magnitude, the altitude error will be correspondingly lower and the optimal gains would be different.

For the case of Figure 17, in which only the improved gains were programmed, we notice a considerable change from Figure 16. Notice how quickly the error builds up to the steady state value; this fast response is due to these gains. The curve of Figure 17 has the same shape of Figure 16, except that it has sharp response features. Also, at time t = 610 seconds, the error drops sharply since the effect of the disturbance terminated at time t = 600 seconds. The altitude error at the time t = 610 seconds is less than 200 feet. It may also be noticed that, while in Figure 17 the error stays at a constant value until the termination of flight at t = 700 seconds, the error in Figure 16 continues to decrease and it appears to go to zero. This, however, is not true. Since the classical gains have an inherent lag, the system is going through a transient at time t = 700 seconds; had the flight duration been extended beyond 700 seconds, the error would have risen again.

In Figure 18, we see a combination of the classical and improved gains. The altitude error follows the pattern of Figure 16 (since the gains are the same) up until time t = 610 seconds. Here, the system switches over to the pattern of Figure 17, and maintains a steady value until time t = 660 seconds, at which instant the classical gains once again take over and the system starts following the pattern of Figure 16 again.

From Figures 16 and 17, at time t = 610 seconds, the error drop corresponds to an improvement of about 70% which is the same as stated earlier in this chapter.

At the same time, the plots for the vertical velocity error for the three flights were also obtained and are as shown in Figures 19, 20 and 21 for the classical, improved and combined gains, respectively. As before, the gains depict a similar behavior. For the classical, the error takes a long time to build up and decrease, whereas for the improved gains, the change is very fast.

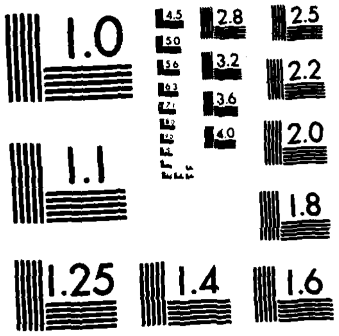
The plots for the baromatric error for the three cases is also shown in Figures 22, 23 and 24. These three plots are exactly the same, thereby confirming the fact that the random number generator was set at the same value at the start of these flights as required.

It was interesting to observe the behavior of the system for one Monte Carlo case out of thirty runs. Figures 25 through 33 show the behavior of the altitude, vertical velocity and barometric errors for the classical, improved and combined gains for one Monte Carlo run, all with the same noise realization.

The behavior of the system is as before, except for one case. Increased noise content is evident in the improved gains. Unfortunately, this behavior is typical; the more the gain K_1 is increased, the more noise content appears in the output. Increasing gain K_1 allows the INS to track the baro altimeter more closely, but because of the noisy contents of the latter, more noise is apt to appear in the output. However, for this case, during the short interval of the TERCOM-update, it may be tolerable.

AD-A124 882 INVESTIGATION OF A THIRD ORDER BARO-DAMPED VERTICAL 2/2
CHANNEL OF INS(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING A RIAZ
UNCLASSIFIED DEC 82 AFIT/GE/EE/82D-55 F/G 17/7 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

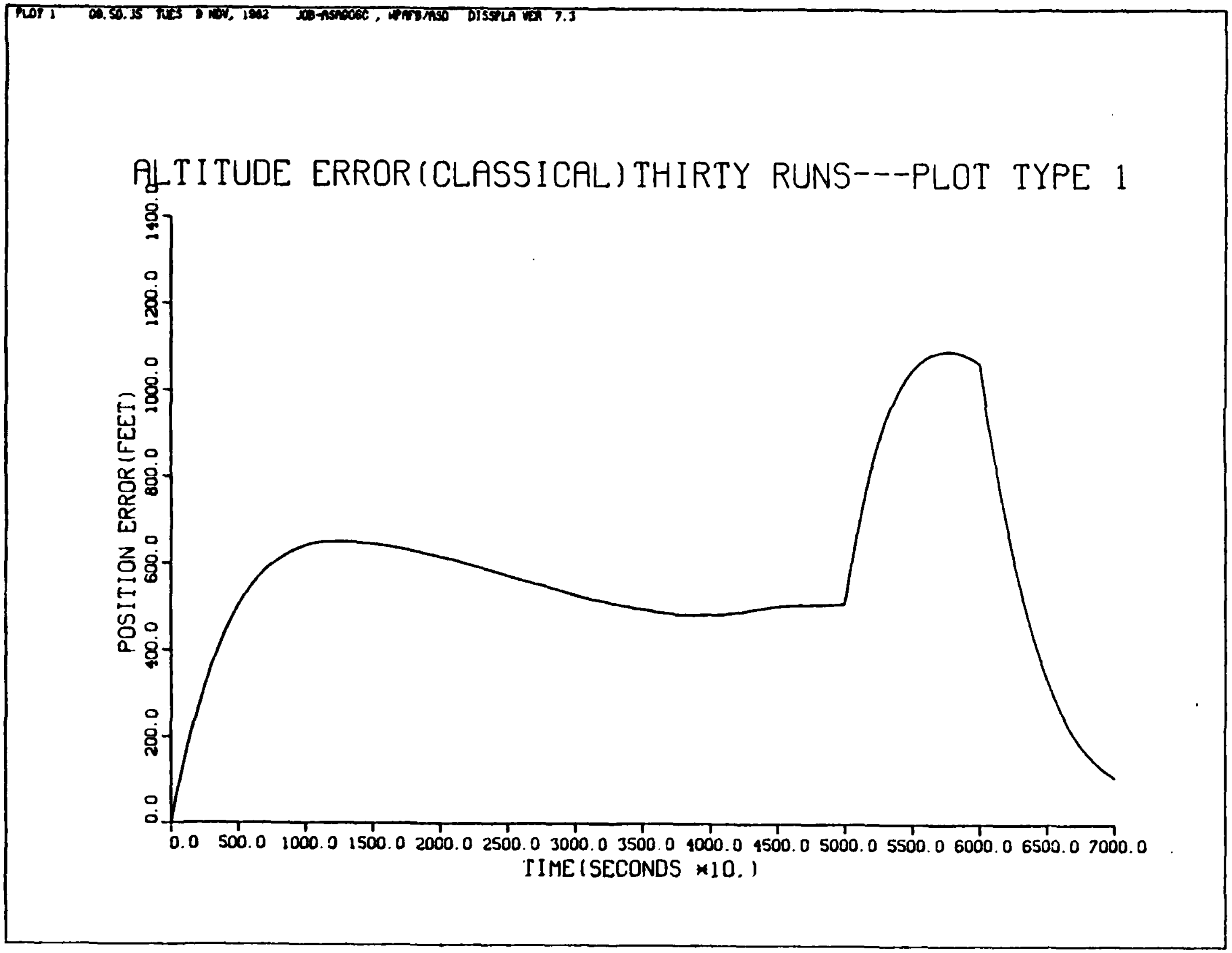


Figure 16. Altitude Error (Classical)

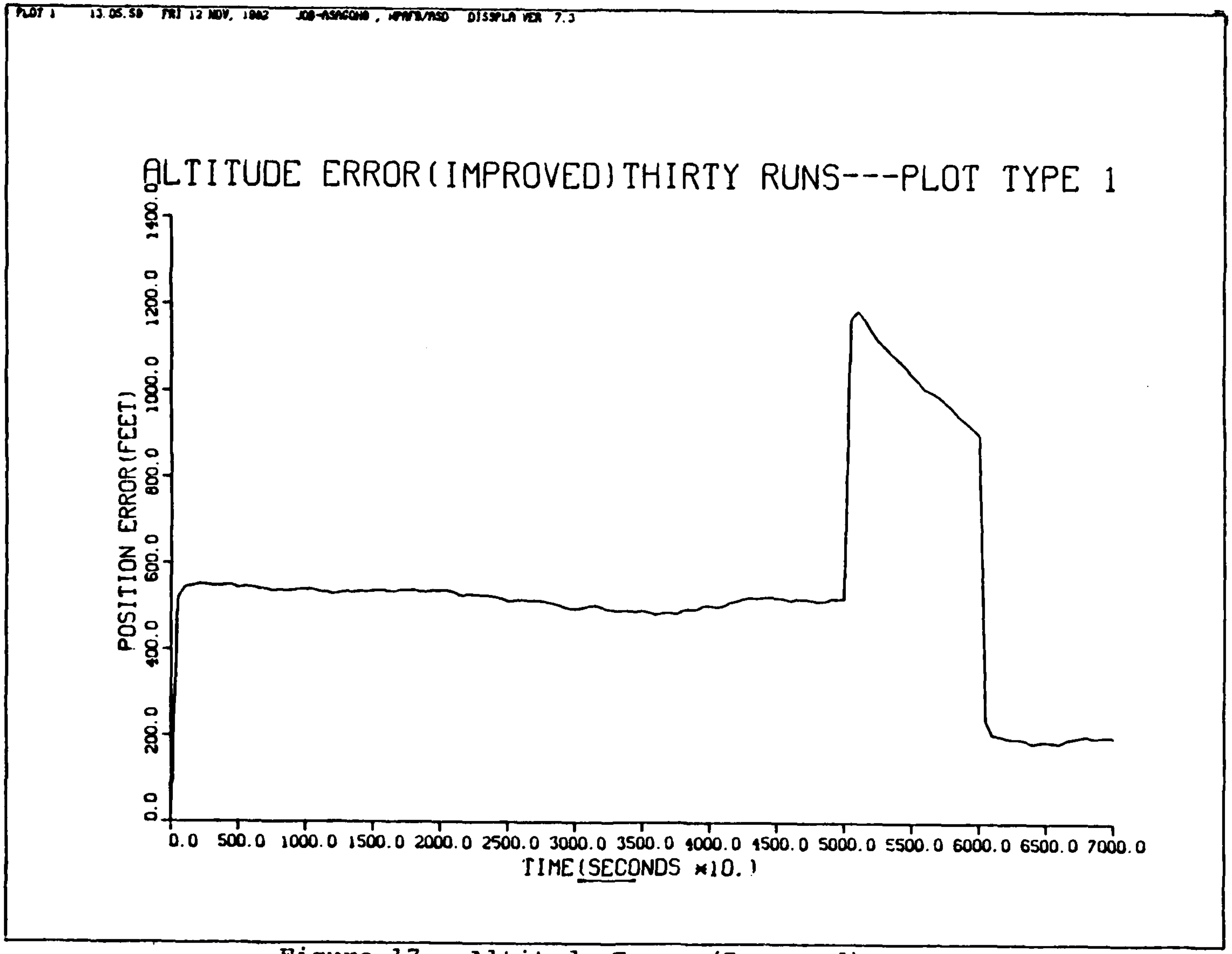


Figure 17. Altitude Error (Improved)

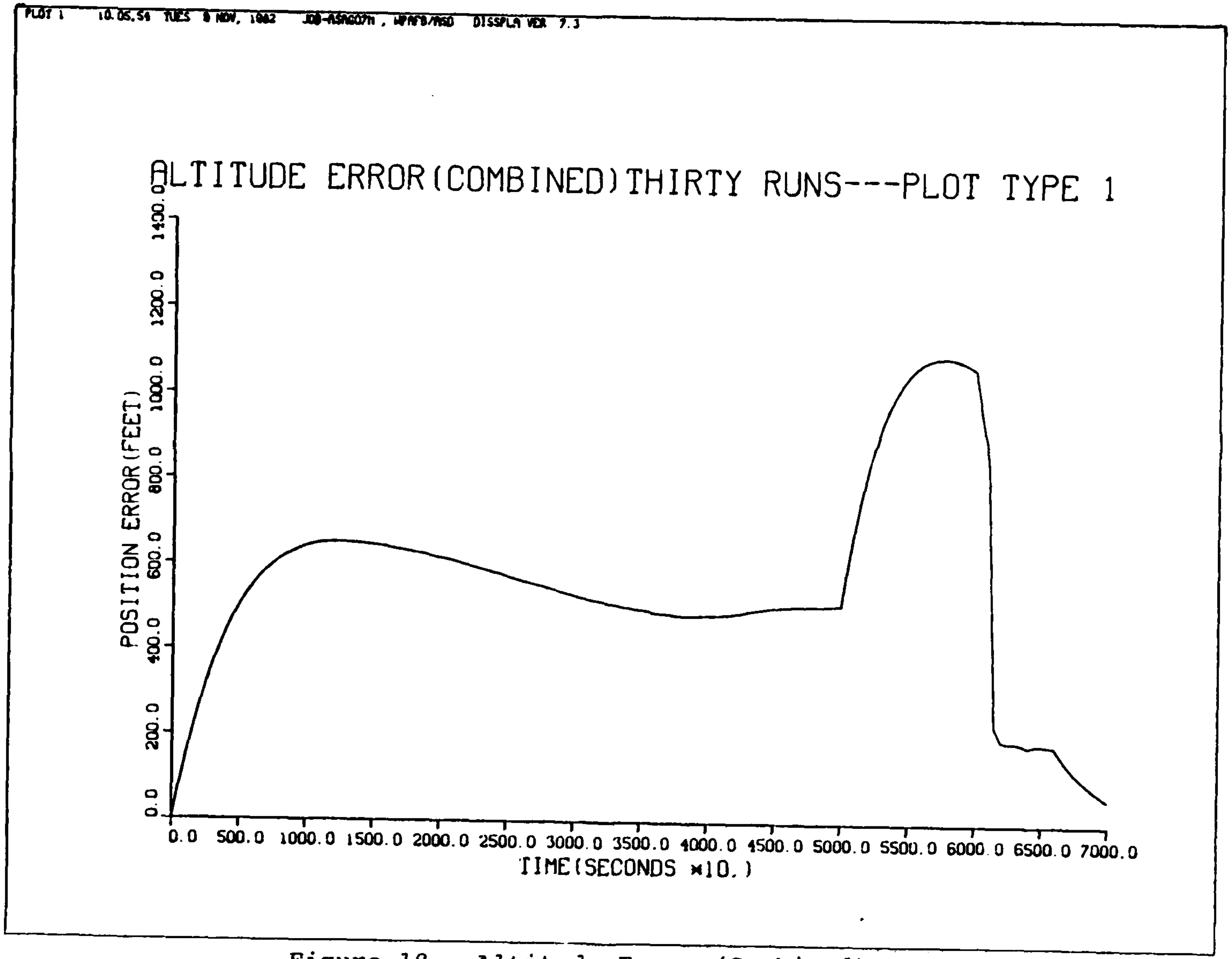


Figure 18. Altitude Error (Combined)

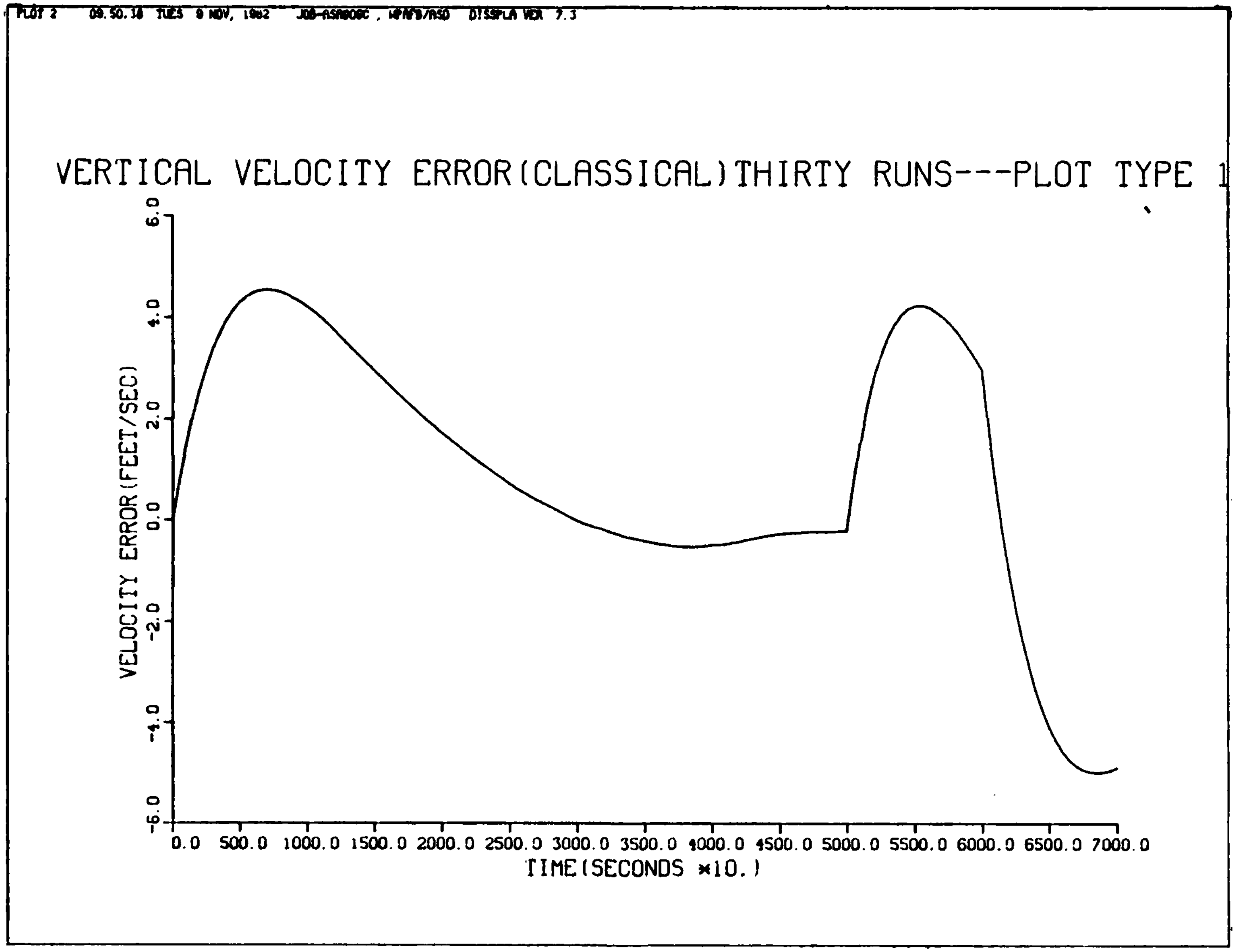


Figure 19. Vertical Velocity Error (Classical)

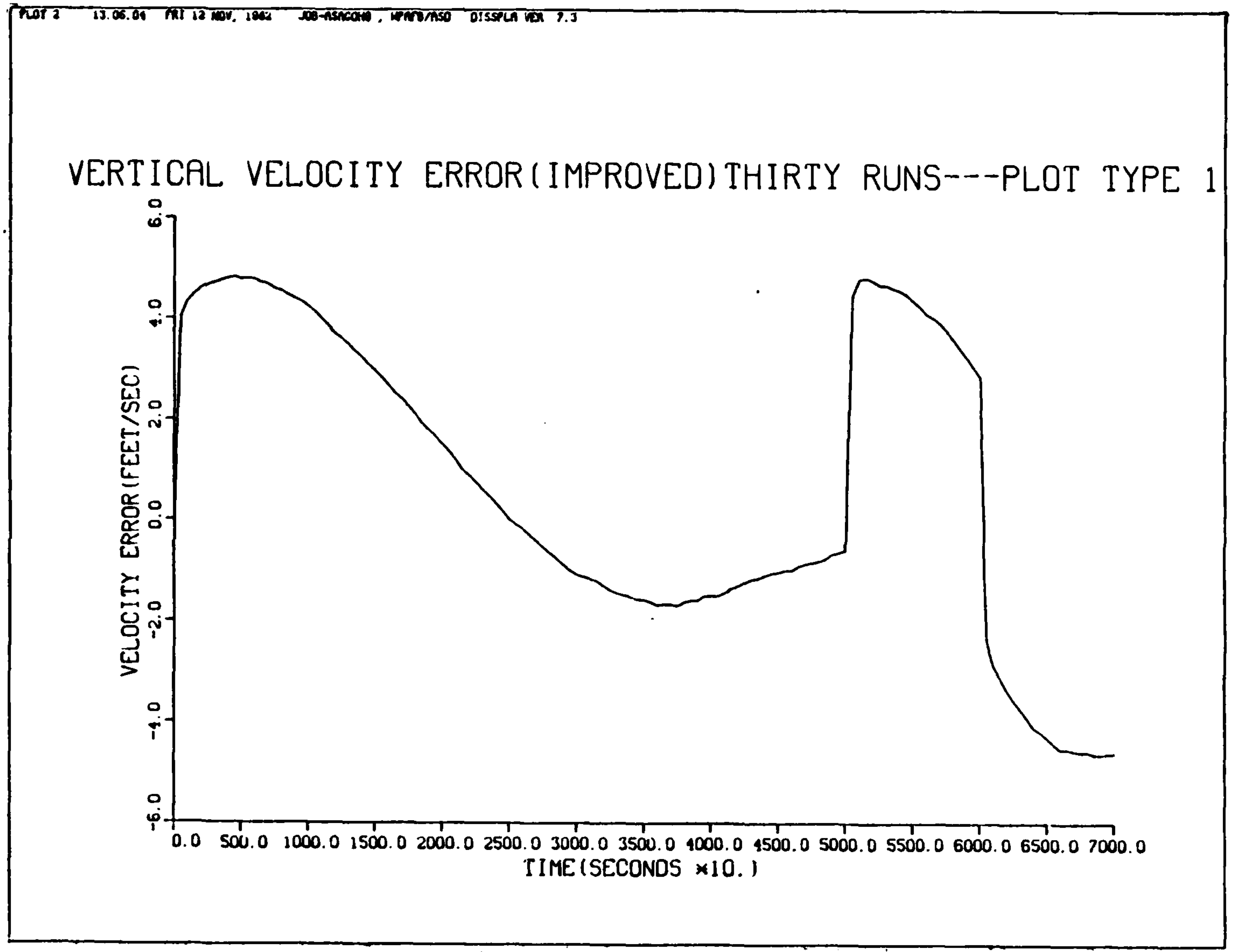


Figure 20. Vertical Velocity Error (Improved)

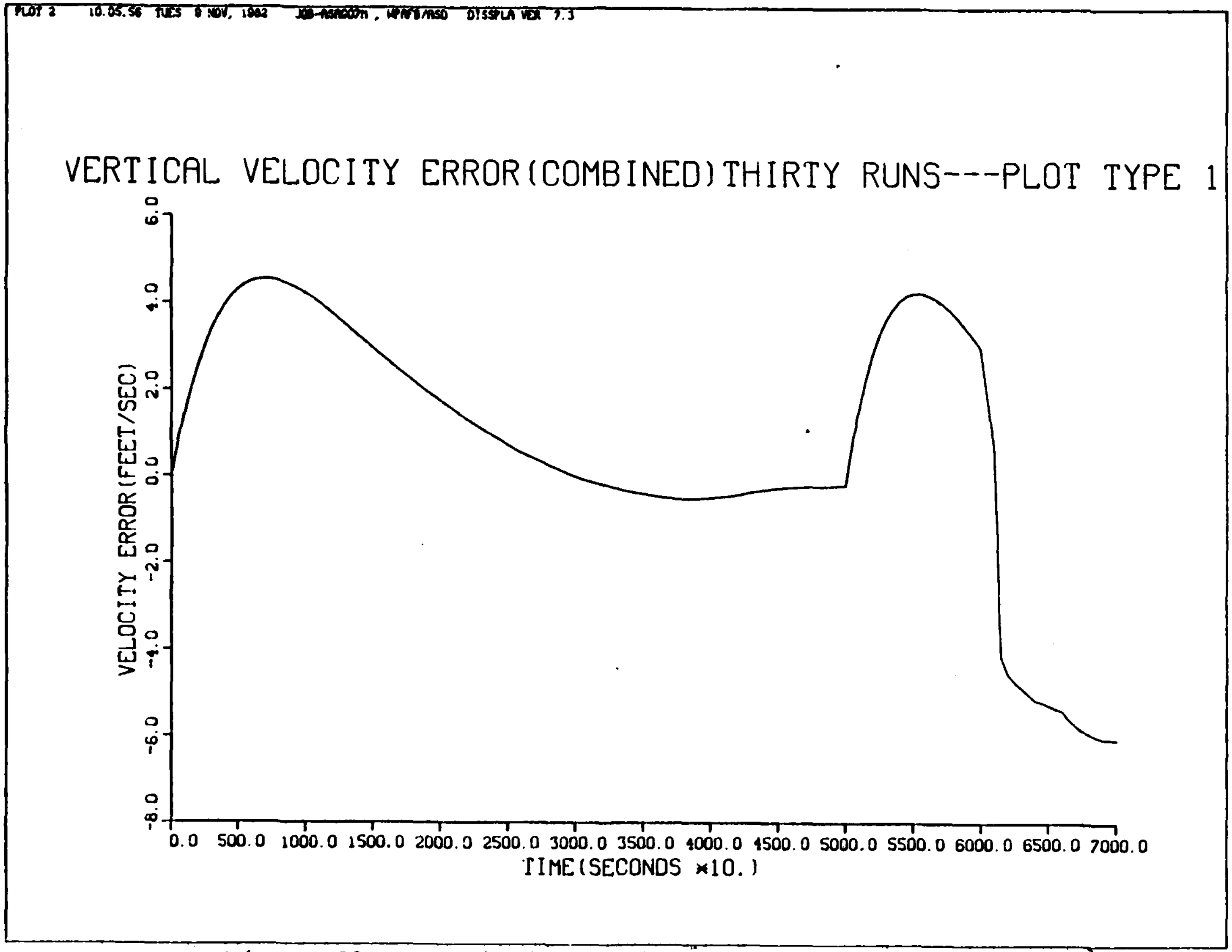


Figure 21. Vertical Velocity (Combined)

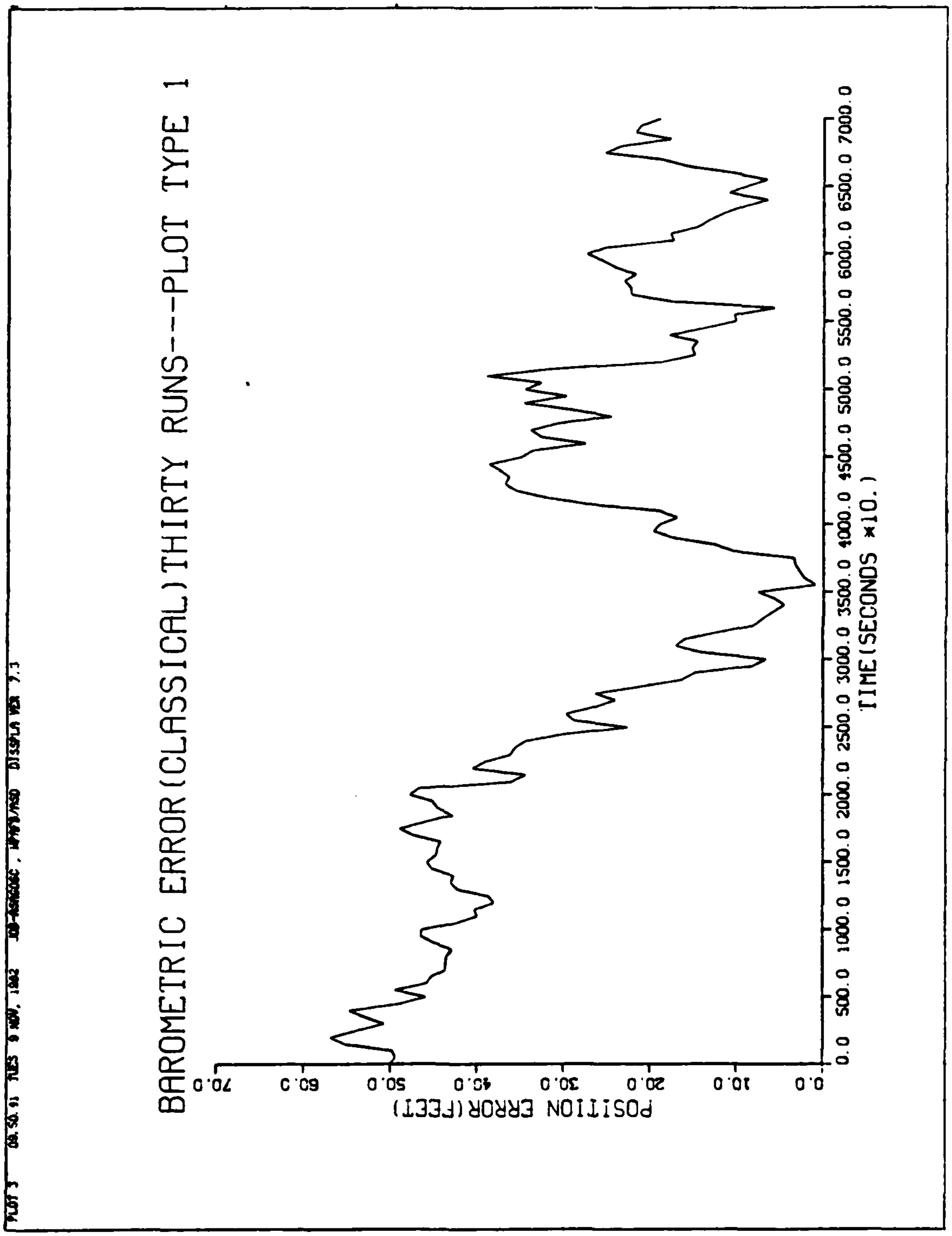


Figure 22. Barometric Error (Classical)

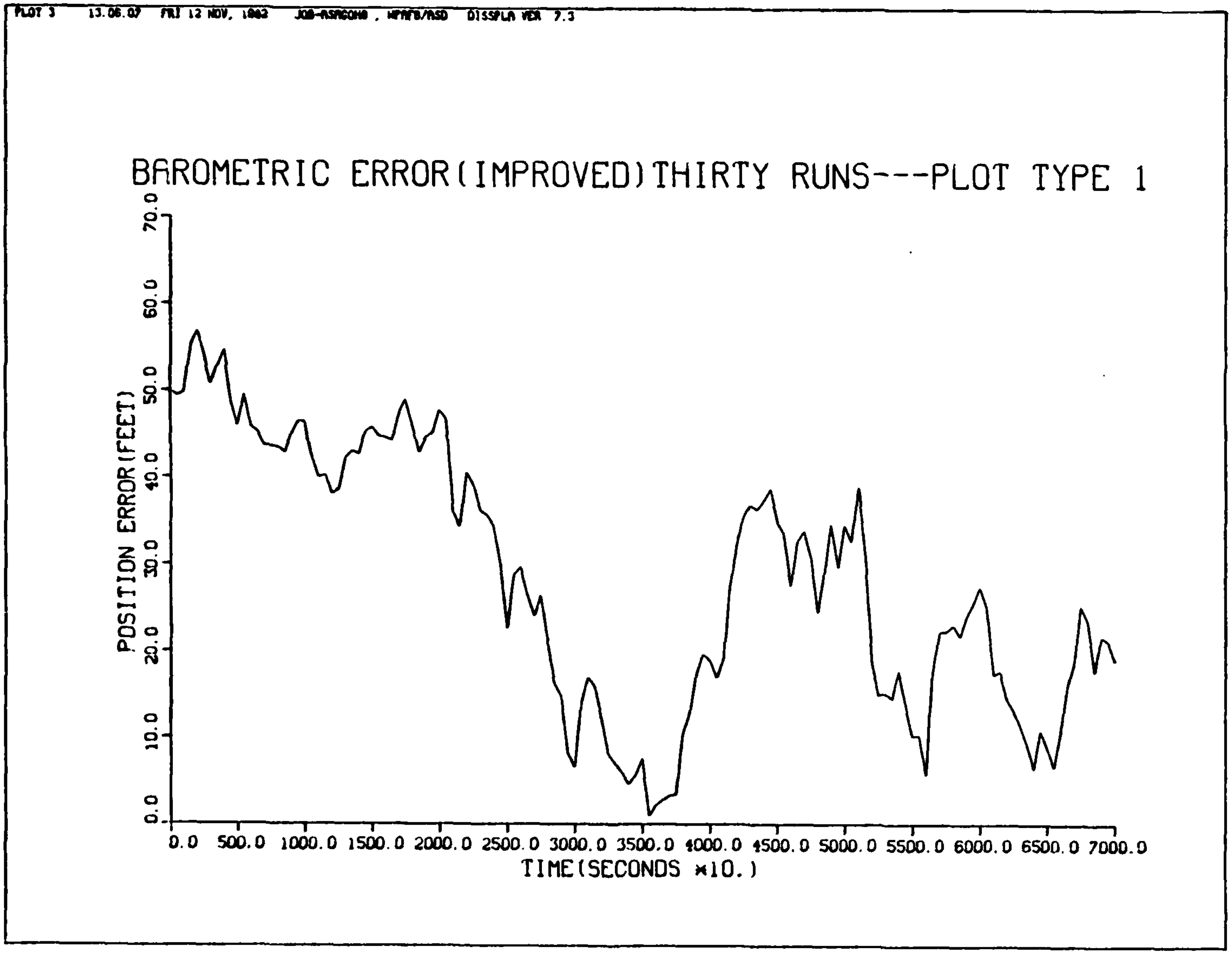


Figure 23. Barometric Error (Improved)

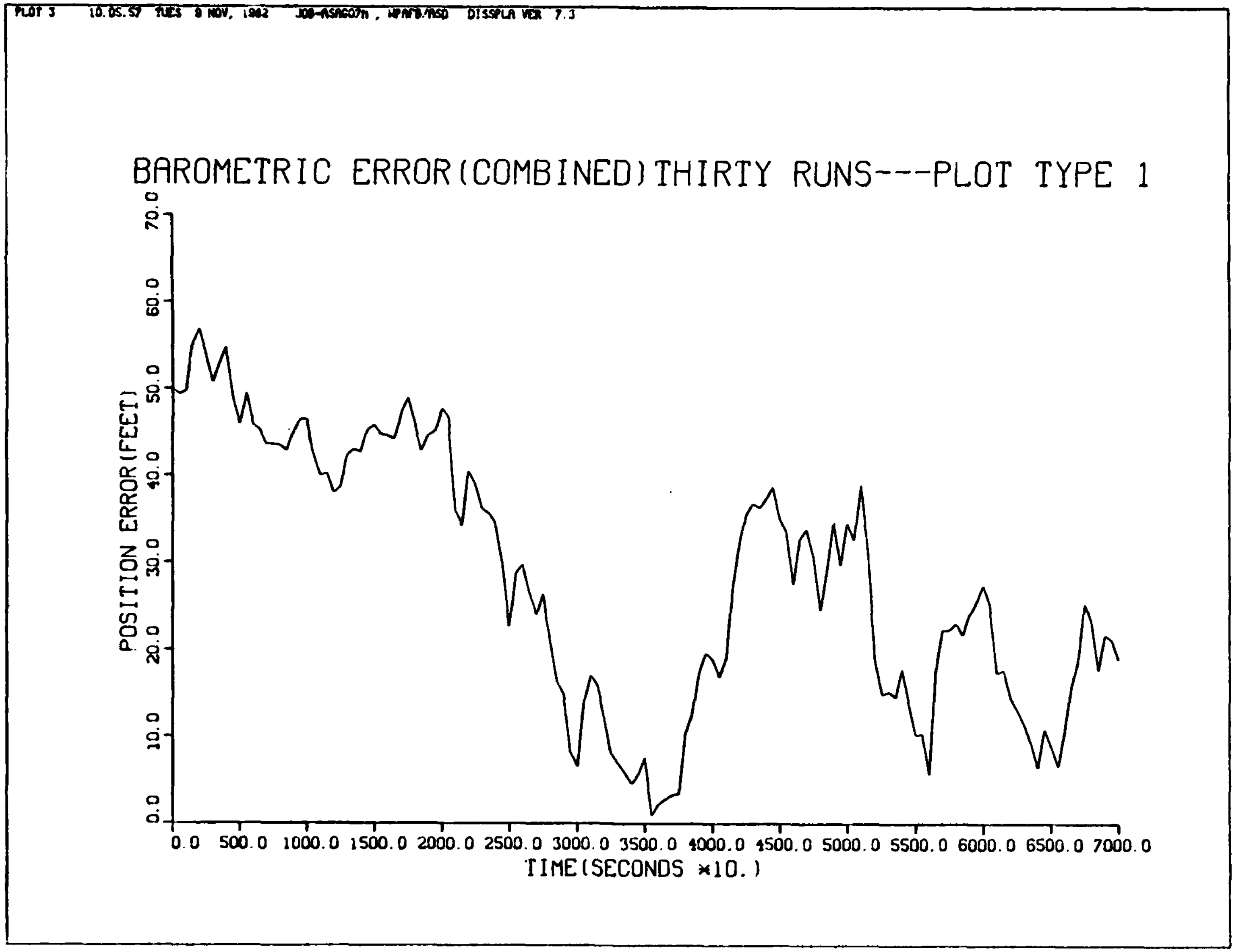


Figure 24. Barometric Error (Combined)

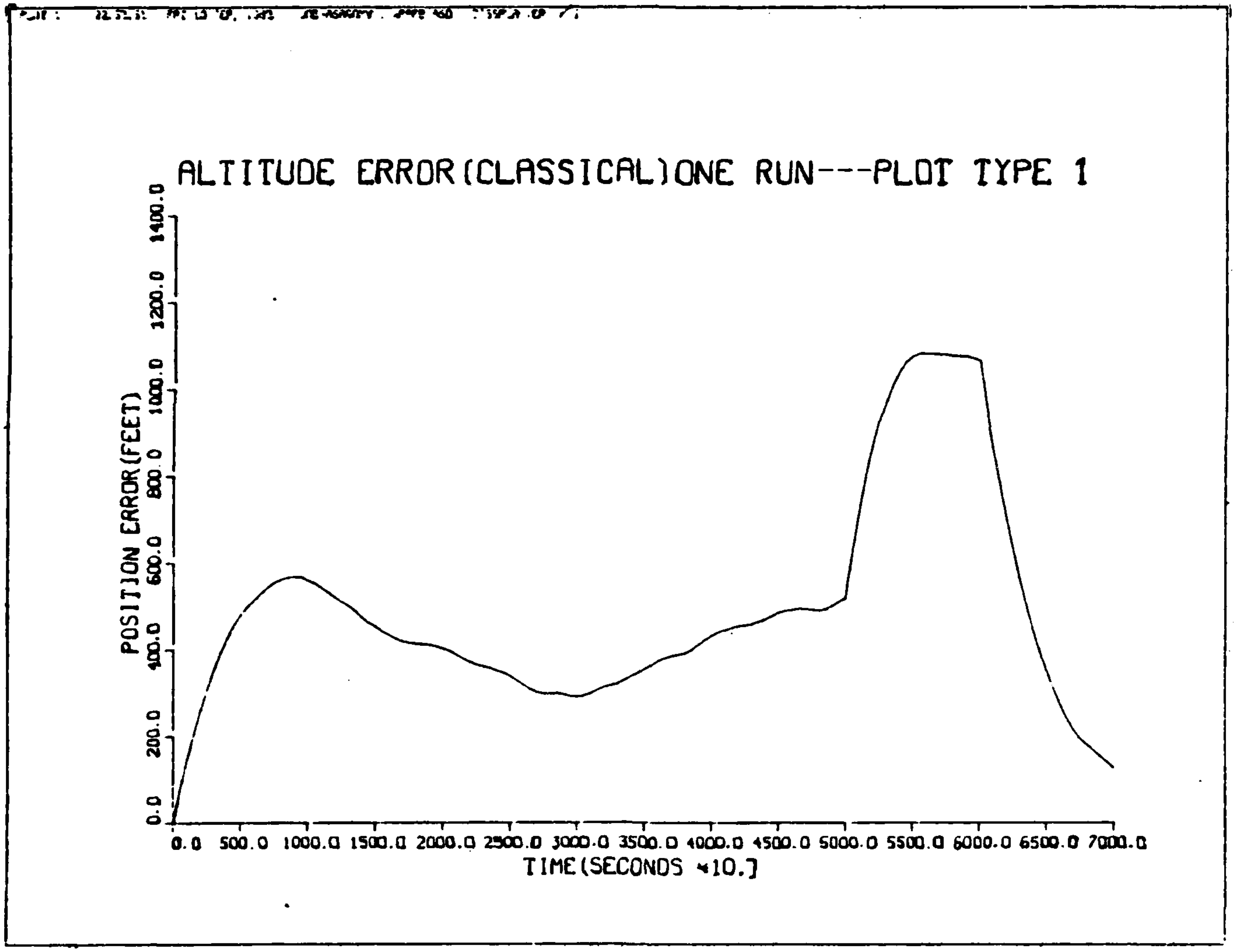


Figure 25. Altitude Error - One Run (Classical)

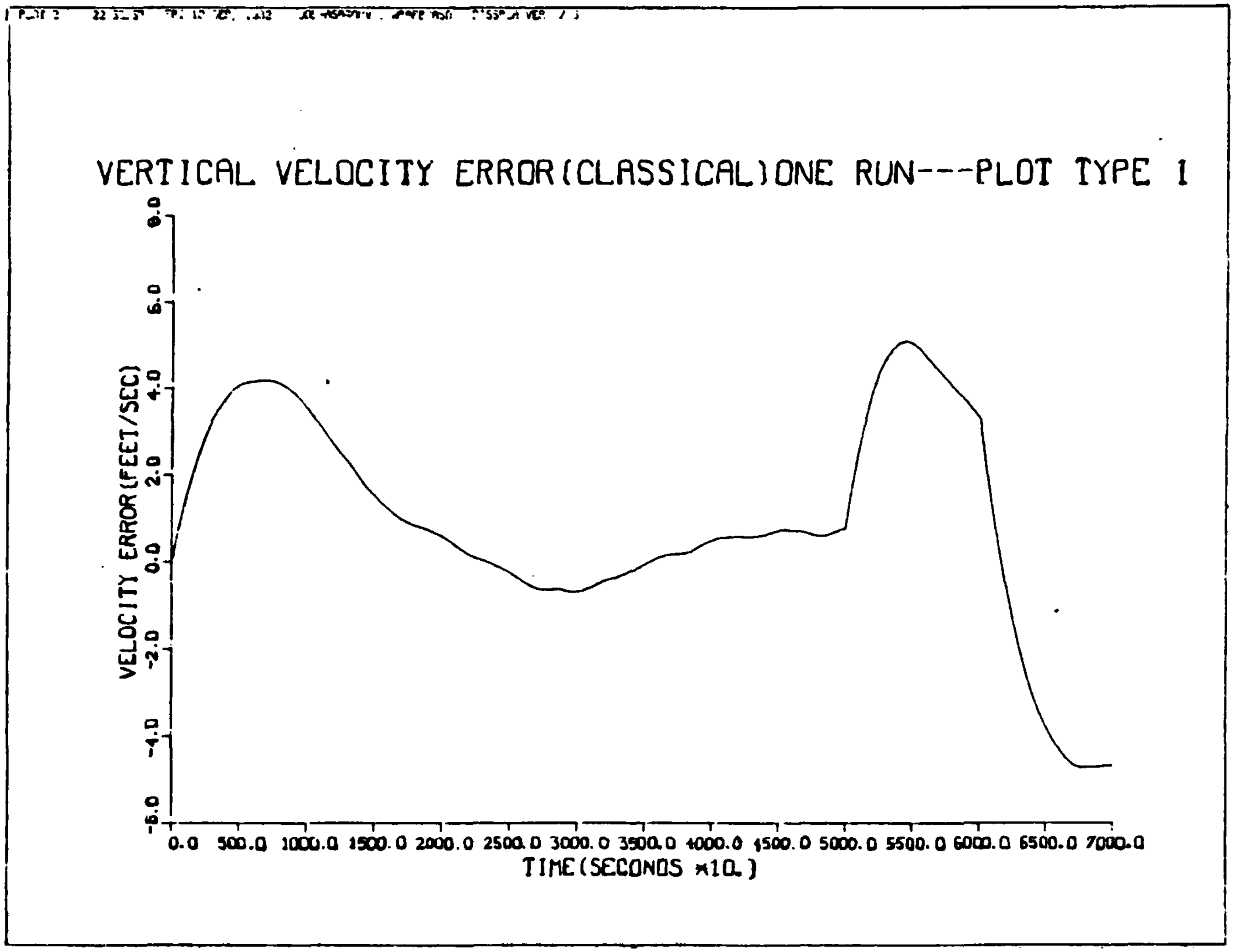


Figure 26. Vertical Velocity Error - One Run (Classical)

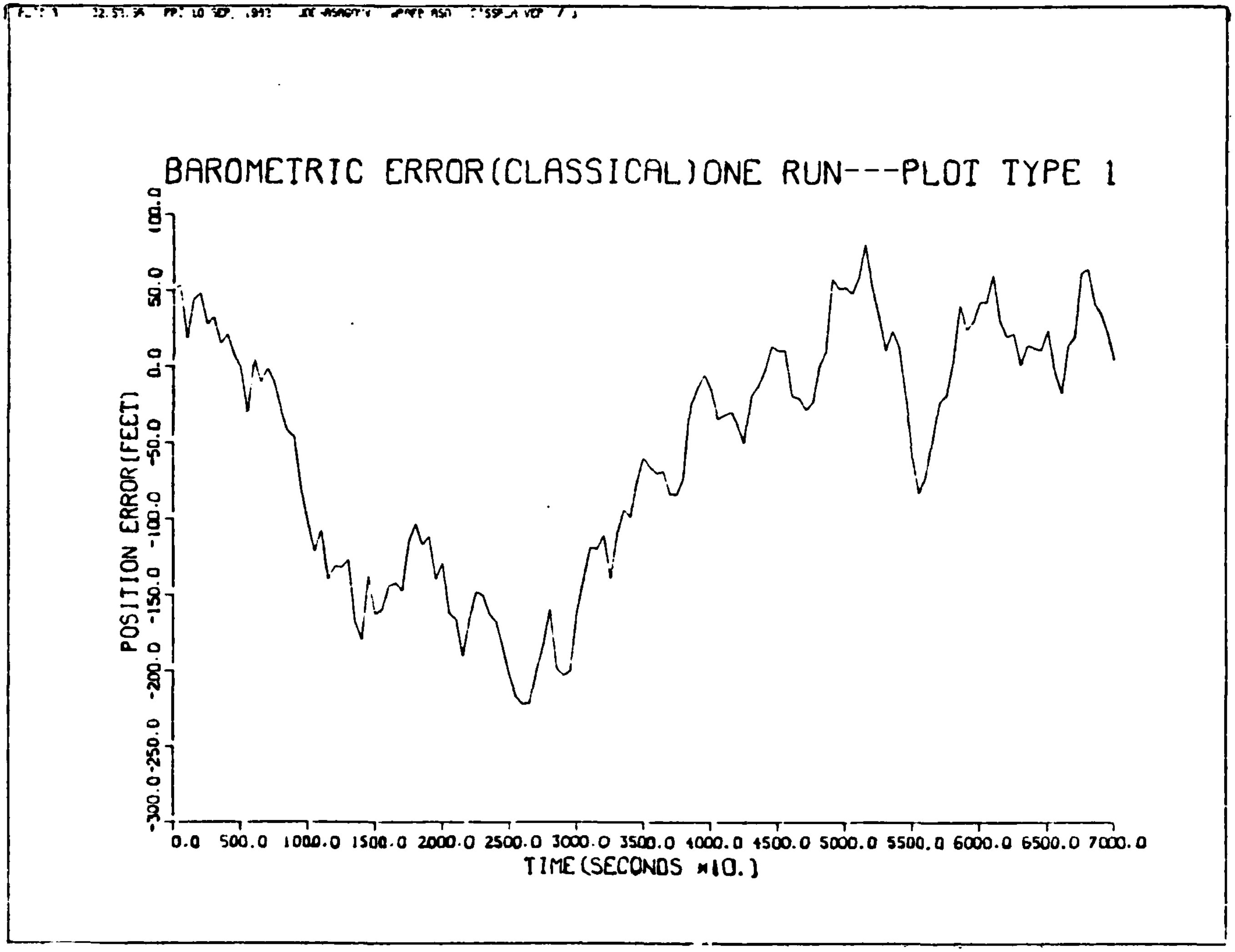


Figure 27. Barometric Error - One Run (Classical)

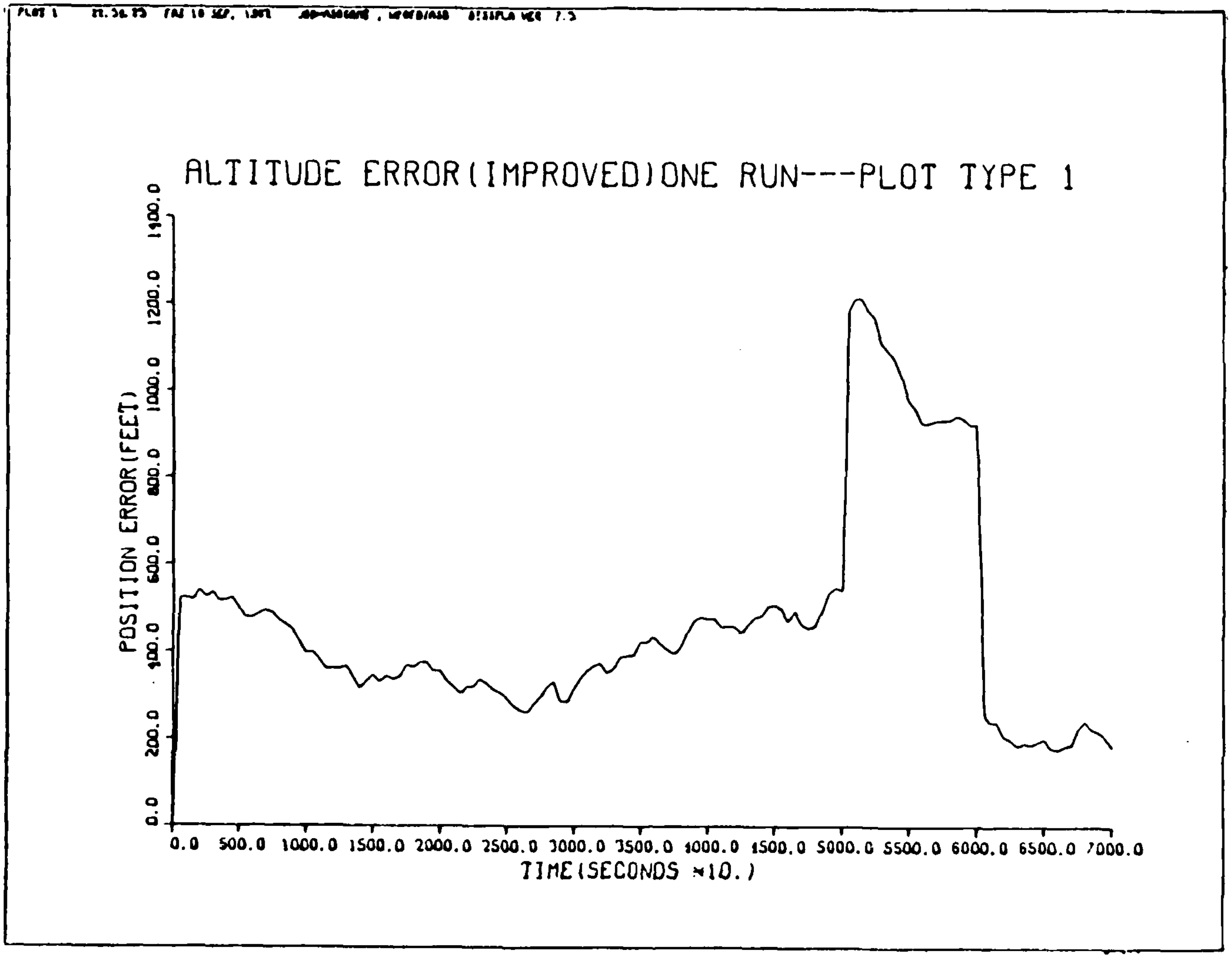


Figure 28. Altitude Error - One Run (Improved)

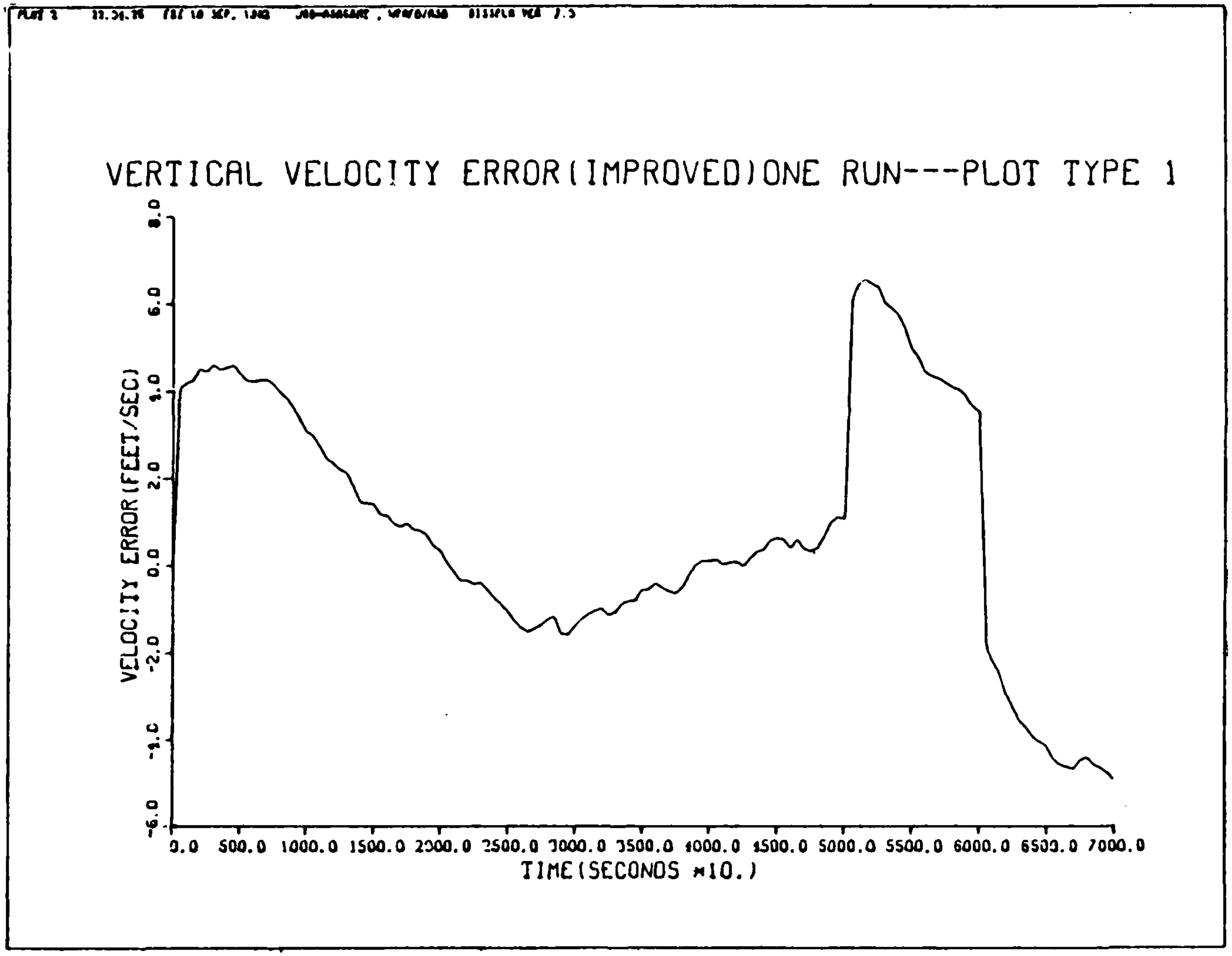


Figure 29. Vertical Velocity Error - One Run (Improved)

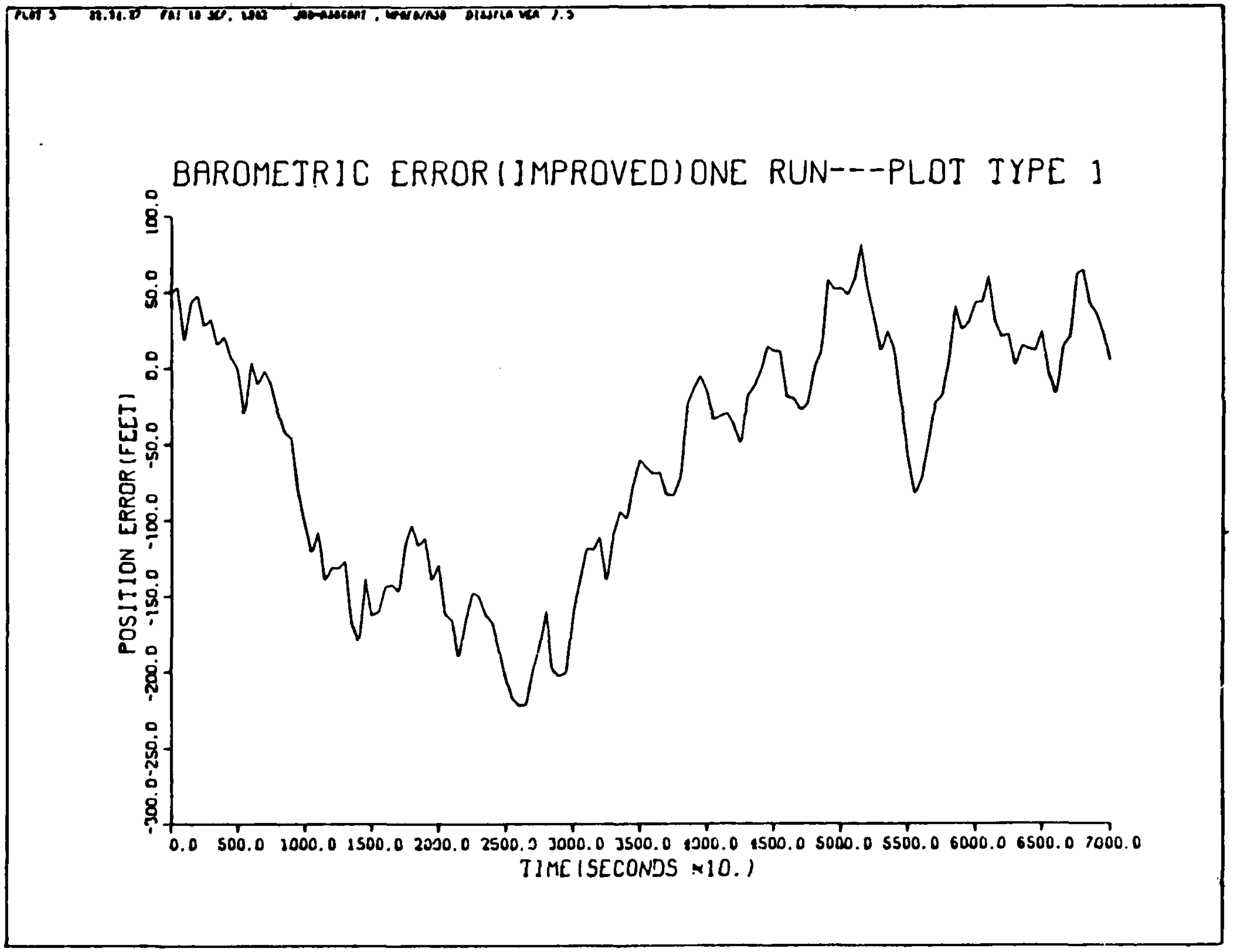


Figure 36. Barometric Error - One Run (Improved)

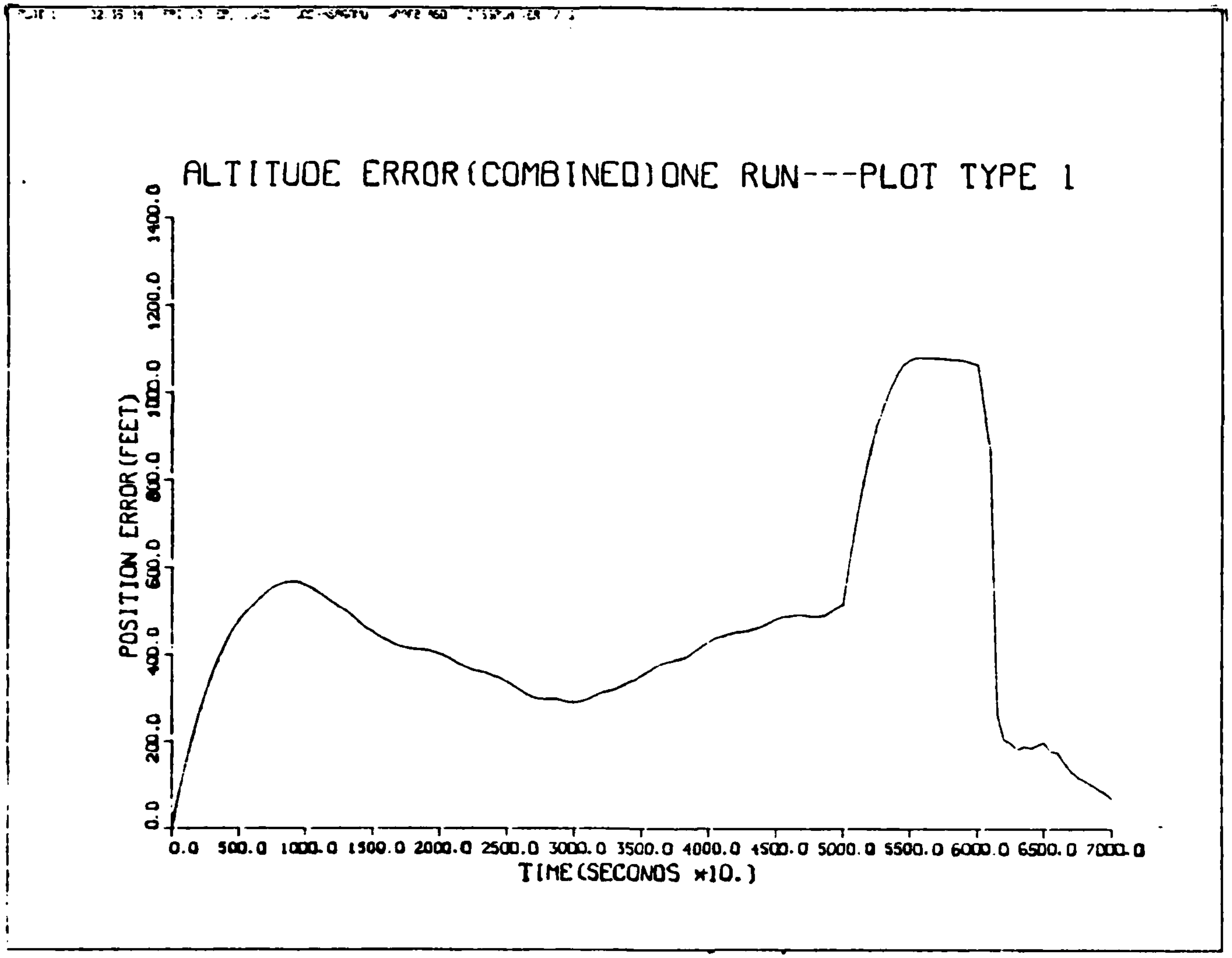


Figure 31. Altitude Error - One Run (Combined)

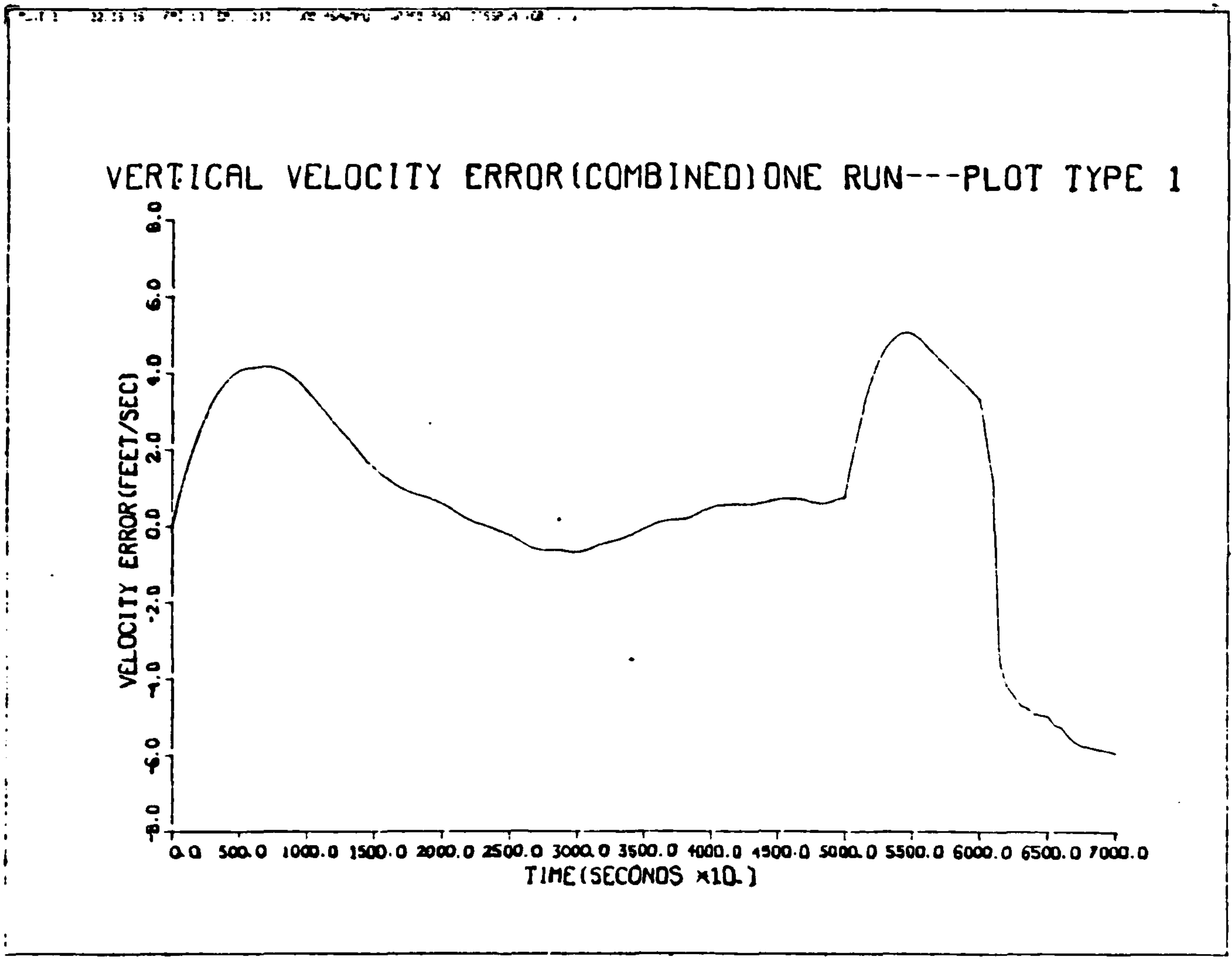


Figure 32. Vertical Velocity Error - One Run (Combined)

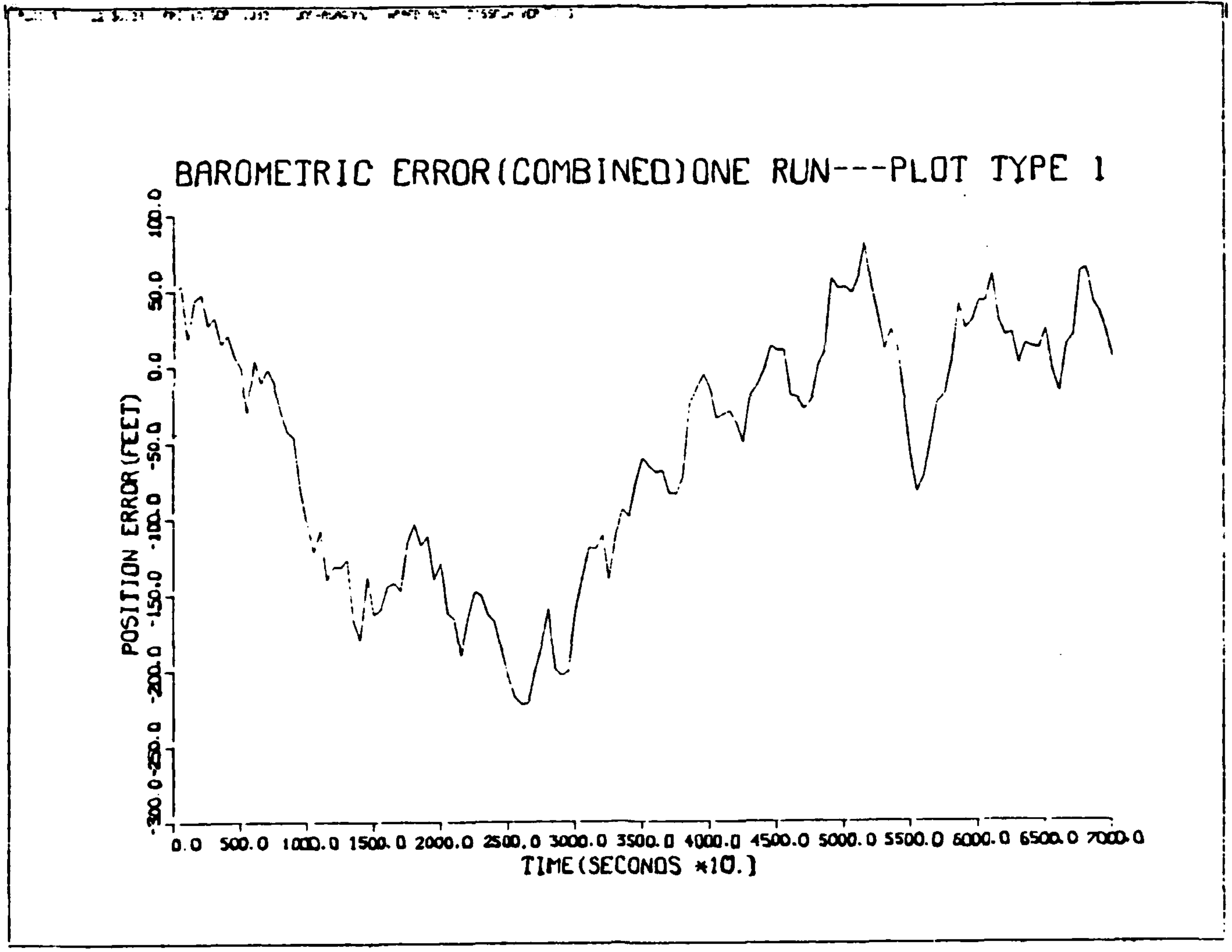


Figure 33. Barometric Error - One Run (Combined)

VI. Conclusions and Recommendations

The simulated flight has demonstrated that using the improved gains for a vehicle carrying out a TERCOM-update, lower mean squared altitude errors are possible as compared with the classical gains. Due to the inherent lag in the classical gains (time constant of 100 seconds), these gains become unsuitable for such a mission. Instead, by optimizing the gains for the period of TERCOM-update, it was shown in the previous chapter, an improvement of 70% was achieved over the classical gains. The classical gains with their long time constant have an advantage, in that errors build up slowly; however, on the other side, the errors also decrease slowly. Thus, the performance achievable through the classical gains is at its best for a level flight of long duration. Any climbs or descents of the vehicle degrade the performance considerably. The optimized gains with a fast time constant showed close tracking capabilities of the vertical channel to the altimeter.

The results of the sensitivity analysis showed that the loop gains are highly dependent on the time intervals during and after the disturbance. Any change of more than 10% on the time intervals $(t_2 - t_1)$ and $(t_4 - t_3)$ would warrant a new set of gains. The results also showed the gains to be highly sensitive to the time interval $(t_3 - t_2)$. Thus, if this thesis needs to be adapted for a different set of time intervals, a different set of optimized gains needs to be searched for (using the search routine). In addition, it was also shown that the gains are a function of the magnitude of the disturbance. Consequently, with a different magnitude of the disturbance, the optimum gains will also be different.

In essence, this thesis has demonstrated that, for a vehicle carrying out a TERCOM-update, it is advantageous to have gains which are radically different from the classical gains. The combination of the improved and classical gains showed that better performance is achievable, rather than with one set of gains only. Although the optimized gains show greater susceptibility to noise than the classical gains, the effect on the system performance is not disturbing due to the short time operation of the former gains.

For the adaptability of this thesis to be such a requirement in the real world environment, it is necessary to define the different time intervals for the TERCOM-update, and then to calculate the gains as done in this paper. The combination of the optimized gains with the classical gains will then provide a lower mean squared altitude error and enable the vehicle to carry out a more accurate TERCOM-update. It is recommended that further study be carried out in determining the optimal combination of the classical and improved set of gains. The time at which the improved set of gains should be switched into the vertical channel needs to be determined, which will give optimum balance between the increased noise level content and optimal performance of the vertical channel in providing lower mean squared altitude errors at TERCOM-update.

Bibliography

1. Draper, C.S., Wrigly, W., and Hovorka, J. *Inertial Guidance*, Pergamon Press, New York, 1960.

2. McClure, C.L. *Theory of Inertial Guidance*, Prentice-Hall, Englewood Cliffs NJ, 1960.

3. Markey, W.R. *The Mechanics of Inertial Position and Heading Indication*, John Wiley and Sons, New York, 1961.

4. Widnall, W.S. and Sinha, P.K. "Optimizing the Gains of Baro-Inertial Vertical Channel," *Journal of Guidance and Control*, *3*: 172-178 (March 1980).

5. Winter, H. *Measurements of Vertical Movements of an Aircraft*, NASA, N76-20152.

6. *Comparison of 3 Vertical Channel Designs for an Integrated GPS/INS*, Intermetrics, Inc., Cambridge MA, December 1977, AFPL-TR-77-208.

7. Widnall, W.S. and Grundy, P.A. *Inertial Navigation System Error Models*, TR-03-73, 11 May 1972 (AD912489L).

8. Maybeck, P.S. *Stochastic Estimation and Control*, Volume I, Academic Press, 1979.

9. Newton, G.C., Gould, L.A., and Kaiser, J.F. *Analytical Design of Linear Feedback Controls*, John Wiley and Sons, New York, 1957.

10. Hindmarsh, A.C. *GEAR: Ordinary Differential Equation System Solver*, Lawrence Livermore Laboratory, Report UCID-30001, Revision 3, December 1974.

11. Fletcher, R. *Fortran Subroutines for Minimization by Quasi-Newton Methods*, Report $7125 AERE, Harwell, England, June 1972.

12. Edwards, R.M. *Gravity Model Evaluation for Precise Terrestrial Inertial Navigation: A System Accuracy Approach*, Technical Report AFAL-TR-79-1231, Avionics Laboratory, Wright-Patterson AFB, December 1979.

13. Musick, S.H. *SOFE: A Generalized Digital Simulation for Optimal Filter Evaluation*, Technical Report AFWAL-TR-80-1108, October 1980, Avionics Laboratory, Wright-Patterson AFB.

14. Musick, S.H., Feldmann, R.E. and Jensen, J.G. SOFEPL: A Plotting Postprocessor for 'SOFE', Avionics Laboratory, Wright-Patterson AFB, November 1981.

15. Broxmeyer, C. Inertial Navigation Systems, McGraw-Hill Book Company, 1964.

Appendix A

Instability of the Vertical Channel

For a vertical accelerometer with input axis along the z-axis in the local-level, the measured specific force is given by

$$f_2 = \ddot{h} + g \tag{A-1}$$

where

f_2 is the specific force

$\ddot{h}$ is the second derivative of the altitude above the earth

g is the acceleration due to gravity.

The gravity in Eq (A-1) can be given by the Taylor series expansion truncated to first order

$$g \simeq g_o - \frac{2g_o}{R_e} h \tag{A-2}$$

where

g_o = gravity at distance r_o

R_e = radius of earth

h = height above earth.

Thus, Eq (A-1) becomes

$$\delta f_2 = \delta \ddot{h} + \delta g \tag{A-3}$$

or

$$\delta \ddot{h} - \frac{2g_o}{R_e} \delta h = \delta f \tag{A-4}$$

Assuming δf to be constant, the solution to Eq (A-4) can be written as

$$h = \frac{\delta f R_e}{2g_o} \left[1 - \cosh \sqrt{\frac{2g_o}{R_e}}\ t \right] \tag{A-5}$$

For a gravity anomaly of 10^{-6} g, the error in altitude after one hour is (Ref 15)

$$\delta h = 28{,}900 \text{ ft}$$

This shows that the vertical channel is unstable for altitude calculations, and external altitude information must be made available to stabilize it.

Appendix B

Minimization Algorithm Listing

```
    THE OBJECT OF THIS PROGRAM IS TO MINIMIZE A COST FUNCTION WITH
    THREE INPUT VARIABLES.IT UTILIZES TWO IMSL ROUTINES NAMELY DGEAR
    AND ZXMIN TO PERFORM THE JOB.THE THREE INPUT VARIABLES TO THE
    ZXMIN ROUTINE ARE THE THREE GAINS OF THE VERTICLE CHANNEL OF THE
    INS.THE VERTICLE CHANNEL IS MODELED BY A SET OF FOUR DIFFERENTIAL
    EQUATIONS WHICH ARE SOLVED BY THE DGEAR ROUTINE.FOR AN INPUT SET
    OF GAINS DGEAR SOLVES FOR THE COST FUNCTION AND ROUTES THE RESULT
    BACK TO ZXMIN WHICH LOOPS A NEW SET OF GAINS ,AND THIS PROCESS
    CONTINUES TILL CRITERIA FOR A MINIMUN IS MET.



    PROGRAM TH8

    EXTERNAL FUNCT
    COMMON/DATA4/H(6)
    DIMENSION G(3),W(9)
    INTEGER MAXFN,N,IOPT
    REAL K(3)


  INITIALIZE INPUT VARIABLES FOR ZXMIN.

    M=3
    NSIG=3
    MAXFN=500
    IOPT=3
    GAIN1=.82

  INITIAL VALUE FOR VERTICAL LOOP GAINS.

    GAIN2=4.91E-3
    GAIN3=5.29E-5
  SET SCALE VALUES FOR LOOP GAINS.

    K(1)=GAIN1/1.E-1
    K(2)=GAIN2*1000.
    K(3)=GAIN3*100000.

  CALL ZXMIN TO MINIMIZE COST FUNCTION.


0   CALL ZXMIN(FUNCT,M,NSIG,MAXFN,IOPT,K,H,G,F,W,IER)
    IF(IER.EQ.0)GO TO 20
  PRINTING ERROR MESSAGES.

    IF(IER.EQ.129)THEN
    PRINT*,'HESSIAN NOT POS. DEF. IER=',IER
    END IF
    IF(IER.EQ.130)THEN
    PRINT*,'IER=',IER
    PRINT*,'MIN. COULD NOT BE ACHIEVED TO NSIG DIGITS'

    REVERSE SCALING
```

```
     PRINT*,'GAIN1=',K(1)*1.E-1
     PRINT*,'GAIN2=',K(2)/1000.
     PRINT*,'GAIN3=',K(3)/100000.
     PRINT*,'COST FUNC=',F
     END IF
     IF(IER.EQ.131)THEN
     PRINT*,'MAXFUN EXCEEDED..IER=',IER
     GO TO 10
     END IF
     STOP

    PRINTING MINIMIZED VALUES OF VERTICAL LOOP GAINS

0    PRINT*,'  '
     PRINT*,'GAIN1=',K(1)*1.E-1
     PRINT*,'GAIN2=',K(2)/1000.
     PRINT*,'GAIN3=',K(3)/100000.
     PRINT*,'COST FUNC=',F
     END

*********************************************************
 THIS SUBROUTINE CALCULATES COST FUNC AND USES DGEAR
 TO SOLVE THE DIFFERENTIAL EQUATIONS.


     SUBROUTINE FUNCT(M,K,F)
     EXTERNAL FCN,FCNJ
     DIMENSION X(4),IWK(4),WK(60)
     COMMON/DATA1/TIMEZ,TIME1,TIME2,TIME3,T
     COMMON/DATA2/DBARO,A11
     COMMON/DATA3/GRAV,RREC,GAIN1,GAIN2,GAIN3
     COMMON/DATA4/H(6)
     COMMON/GEAR/DUMMY(48),SDUMMY(4),IDUMMY(38)
     REAL K(M),J1,J2

   SET FIRST ORDER LAG CONSTANT
     A=5.7931605E-4

   SET SIGMA VALUE FOR 1ST ORDER MARKOV PROCESS(METERS)
     SIGMA=152.4

   SET NOISE VALUES.
     QA1=2.4E-4
     QA2=1.E-9
     QB1=100.
     QB2=2*A*(SIGMA**2)
     N=4

   SET INITIAL VALUES FOR INTEGRATION
     T=0.0
     X(1)=0
     X(2)=0
     X(3)=0
     X(4)=0
     TOL=2.E-10
     S=.000001
     METH=2
     MITER=2
     INDEX=1
```

```
  TSTOP=160.

WRITE VALUES FOR PLOTTING ON TAPE 5
  REWIND 5
  WRITE(5) T,X
  TEND=0.
  ICOUNT=0

GRAVITY IN METERS/SEC**2
  GRAV=9.80665

RADIUS OF EARTH IN METERS.
  R=6378165.0
  RREC=1./R


REVERSE SCALING

  GAIN1=ABS(K(1))*1.E-1
  GAIN2=ABS(K(2))/1000.
  GAIN3=ABS(K(3))/100000.

SET TIME VARIABLES

  TIMEZ=0.
  TIME1=100.
  TIME2=110.
  TIME3=160.

  BETA=.5

SET DISTURBANCE MAGNITUDE(SQUARED)METERS

  ALPHA=200**2


  C=2*GRAV*RREC
  CALL UPDATE

REINITIALIZE DGEAR FOR STEP INPUTS
  IF(T.EQ.100)THEN
  INDEX=1
  S=1.E-6
  END IF
  IF(T.EQ.110)THEN
  INDEX=1
  S=1.E-6
  END IF

  IF(T.LT.TSTOP)THEN
  ICOUNT=ICOUNT+1
  TEND=T+5.

CALL DGEAR
```

```
0     CALL DGEAR(N,FCN,FCNJ,T,S,X,TEND,TOL,METH,
     +MITER,INDEX,IWK,WK,IER)

     PRINTING ERROR MESSAGES

      IF(IER.GT.128)THEN
      PRINT*,IER
      IF(IER.EQ.132)THEN
      DO 30 I=1,N
      CD=WK(IDUMMY(11)+I)/WK(I)
      PRINT*,CD
0     CONTINUE
      END IF
      STOP

      END IF
      WRITE(5) T,X
      GO TO 5
      END IF

     COMPUTE COST DUE TO DISTURBANCE
      J1=ALPHA*X(4)

     COMPUTE VARIABLES FOR J2 COST FUNC
      B1=QA1/(2*((GAIN1)*(GAIN2-C)-GAIN3))
      B2=(GAIN1*QA2)/(2*((GAIN1*GAIN3)*(GAIN2-C)-(GAIN3**2)))
      B3N=(GAIN1**2)*(GAIN2-C)
      B3N=B3N+(GAIN2**2)-(GAIN1*GAIN3)
      B3D=2*((GAIN2-C)*GAIN1-GAIN3)
      B3=B3N/B3D
      B3=B3*QB1
      B4N=(GAIN1*GAIN2)*((A**2)*GAIN1+A*GAIN2+GAIN3)
      B4N=B4N-GAIN1*((A**2)*C*GAIN1+(A**2)*GAIN3+C*GAIN3)
      B4N=B4N+(A*GAIN2)**2-GAIN3**2
      B4D=A*GAIN1*GAIN2*((A**2)+A*GAIN1+GAIN2-2*C)
      B4D=B4D+A*C*GAIN1*(C-(A**2)-A*GAIN1)
      B4D=B4D-GAIN1*GAIN3*(C+(A**2))
      B4D=B4D+GAIN3*(A*C-A*A*A-GAIN3)
      B4D=B4D+GAIN2*GAIN3*(GAIN1-A)
      B4D=2*A*B4D
      B4=B4N/B4D
      B4=B4*QB2
```

```
 COMPUTE COST DUE TO NOISE INPUTS
   J2=(TIME3-TIME2)*(B1+B2+B3+B4)

 COMPUTE TOTAL COST
   F=(BETA*J1)+(1-BETA)*J2

 PRINT VARIABLES ITERATIVELY
   PRINT*,'GAIN1=',GAIN1
   PRINT*,'GAIN2=',GAIN2
   PRINT*,'GAIN3=',GAIN3
   PRINT*,'B1=',B1
   PRINT*,'B2=',B2
   PRINT*,'B3=',B3
   PRINT*,'B4=',B4
   PRINT*,'J1=',J1
   PRINT*,'J2=',J2
   PRINT*,'COST FUNC=',F
   PRINT*,'  '
   RETURN
   END
*************************************************************
 THIS SUBROUTINE CALCULATES SETS THE DISTURBANCE INTERVAL

   SUBROUTINE UPDATE
   COMMON/DATA1/TIMEZ,TIME1,TIME2,TIME3,T
   COMMON/DATA2/DBARO,A11
   IF(T.GE.TIMEZ.AND.T.LE.TIME1)THEN
   DBARO=1
   A11=0.
   ELSE IF(T.GE.TIME2.AND.T.LE.TIME3)THEN
   DBARO=0.
   A11=1.
   ELSE
   DBARO=0.
   A11=0.
   END IF
   RETURN
   END
*************************************************************

 THIS SUBROUTINE CALCULATES DERIVATIVES FOR DGEAR

   SUBROUTINE FCN(N,T,X,XDOT)
   COMMON/DATA2/DBARO,A11
   COMMON/DATA3/GRAV,RREC,GAIN1,GAIN2,GAIN3
   DIMENSION X(N),XDOT(N)
   XDOT(1)=X(2)-((GAIN1)*(X(1)-DBARO))
   XDOT(2)=((2*GRAV*RREC)-GAIN2)*X(1)+(GAIN2*DBARO)-X(3)
   XDOT(3)=(GAIN3*X(1))-(GAIN3*DBARO)
   XDOT(4)=A11*(X(1)**2)
   RETURN
   END
*************************************************************

 THIS SUBROUTINE ACTS AS A DUMMY FOR DGEAR
   SUBROUTINE FCNJ(N,T,X,PD)
   INTEGER N
   REAL X(N),PD(N,N),T
   RETURN
   END
..
```

Appendix C

SOFE User Input Routines

Introduction

SOFE (Ref 13) is a Monte Carlo simulation program that helps in analyzing integrated systems employing Kalman Filter estimation techniques. It can also be used for propagating the navigation error equations over a desired trajectory.

SOFE requires both truth and filter model state variables. Since this thesis had only truth model states, a dummy filter state was introduced to satisfy the SOFE requirements. Brief discussions on each of the input routines used in this thesis are described in the following paragraphs. For additional information, Reference 13 can be consulted.

Tape 5

This file is the input to SOFE for the following information:

(a) Problem title

(b) PRDATA information for initializing matrices and time intervals in basic SOFE

(c) Initial values for the truth model error variables

(d) Input for the USRIN routine

(e) Plotting information

Tape 5 input can reside on card decks or can be entered interactively.

Subroutine AMEND

This subroutine is used to apply total feedback control after a certain number of measurement intervals. Since no filter state was used, this routine is just a stub.

Subroutine ERDY

This subroutine was generated to calculate the non-zero entries of the 9x9 fundamental matrix of the INS differential equations. This subroutine is not explicitly required by SOFE, and it was generated for the purpose of program clarity.

Subroutine ESTIX

This subroutine is required for computing the user defined quantities. Since no quantities were required to be computed, this routine was also a stub.

Subroutine FQGEN

This subroutine is required for the filter states which were not present in this thesis. Thus, this routine was also a stub.

Subroutine HRZ

This routine is again required for the filter states and thus it was also a stub.

Subroutine NUUNIT

This routine was generated to convert the statistical input to computational units. Thus, all units conversion calculations were done here.

Subroutine SNOYS

This user routine adds gaussian random samples to specified truth states to simulate the accumulated effect of process driving noise on these states over the noise accumulation interval.

Subroutine STABLE

This subroutine was generated for printing a table of the statistical input for the truth model. This routine is not required for SOFE explicitly; however, it helps in fault analysis.

Subroutine TRAJ

Since no external trajectory program was used, this routine was generated for establishing the great circle flight trajectory.

Subroutine USRIN

This user defined routine was used for reading and printing input data.

Subroutine XFDOT

This routine is required for the filter states, and therefore it was just a stub.

Subroutine XSDOT

This subroutine contains the derivatives of the truth model and it also initializes the various error states.

TAPE 5

```
100=LN15/VERTICAL CHANNEL ERROR PROPAGATION----CLASSICAL VS IMPROVED
110= $PRDATA
120= NF=1,NS=50,M=0,NZF=0,NXTJ=9,
130= LXTJ=.F.,
140= TO=0.,  TF=700.,
150= DTPRNT=5.,   DTCCPL=5.,
160=              DTNOYS=3.,
170= DTPRPL=10.,
180= LPRXF=.F.,  LPRDG=.F.,
190= IPRRUN=1,   IPGSIZ=55,
200= LPF=.T.,  LCC=.T.,
210= TOLER=.0001, HMAX=60.,HMIN=.0001,
220= ISEED=-2361268, IPASS=30,
230= $
240= 50*0.
250=0.
260=1,1,1.
270=0,0,0.
280= $SIGOS  DYNAMIC(1)=10*0.,
290= GB(1)=2*.0005,.0007,GF(1)=6*.0005,GFF(1)=3*0.,
300= GSF(1)=3*5.,GM(1)=6*2.5,
310= AB(1)=3*25.,ASF(1)=3*25.,AM(1)=6*5.15,
320= BARO=50.,GDE=26.,GND=17.,GA=35.,
330= BASF=.03,SPC=1.54E-04,AL=.25,
340=$
350= $STATS   DALTS=250.,DGRAVE=10.,DGRAVN=10.,
360= DGA=60.,GYNDS(1)=3*0.0,ACNDS(1)=3*0.0,
370= BASIGS=250.,EDSIGS=26.,DNSIGS=17.,GASIGS=35.,
380= $
390= $CONTRL   LFDBK=.F.,LINTOO=.F.,KXS60=1,
400= $
410= 1.0
420= 3,0,0,1.
430= 6,0,0,1.
440=44,0,0,1.
450= 0,0,0,0.
460=   TIME(SEC)
470=ALTITUDE ERROR
480=POSITION ERROR *FEET*
490=VERTICAL VELOCITY ERROR
500=VELOCITY ERROR *FPS*
510=BARO ALTIMETER ERROR
520=ALTITUDE *FEET*
```

```
 1                SUBROUTINE AMEND(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ)
            C
            C  USER-WRITTEN SUBROUTINE TO APPLY TOTAL FEEDBACK CONTROL
            C
 5                COMMON /CNTRL/LFDBK,LINT00,KXS30
                  DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ)
            C
                  LOGICAL LFDBK
            C
10                RETURN
                  IF (.NOT. LFDBK) RETURN
            C
                  DO 100 I=1,9
                     XS(I) = XS(I) - XF(I)
15                   XF(I) = 0.
            100   CONTINUE
                  DO 200 I=11,13
                     XS(I) = XS(I) - XF(I)
                     XF(I) = 0.
20          200   CONTINUE
                  XS(44) = XS(44) - XF(10)
                  XS(45) = XS(45) - XF(14)
                  XS(49) = XS(49) - XF(15)
                  XF(10) = 0.
25                XF(14) = 0.
                  XF(15) = 0.
                  RETURN
            C
            C  ***************************************************************
30                ENTRY AMENDC
                  RETURN
                  END


CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

     11    I                  THERE IS NO PATH TO THIS STATEMENT.
```

```
SUBROUTINE ERDY       74/74    OPT=1                           FTN 4.8+564         11/03/82  15.33.

      SUBROUTINE ERDY(NTRAJ,TRAJ,F9X9)
C
C  ERDY COMPUTES NON-ZERO ENTRIES FOR THE 9X9 SUBBLOCK THAT
C  MAKES UP THE FUNDAMENTAL NAVIGATION ERROR DYNAMICS MATRIX.
C  ERDY ALSO FILLS COMMON AREA 'SHARE' WITH CURRENT VALUES OF
C  SEVERAL COMPUTED VARIABLES, INCLUDING X-Y-Z VELOCITIES,
C  SPECIFIC FORCES AND COMMANDED ANGULAR VELOCITIES, PLUS
C  WANDER-ANGLE VALUES.  'SHARE' VARIABLES ARE USED IN XSDOT,
C  XFDOT AND FQGEN FOR A VARIETY OF STATE AND COVARIANCE
C  COMPUTATIONS.
C
      COMMON /EARTH/OMEGA,REQ,ESQ,GEE
      COMMON /MATH/QTRPI,HALFPI,PI,TWOPI,RPD
      COMMON /VDAMP/CK1,CK2,CK3
      COMMON /SHARE/VX,VY,VZ,VG,FX,FY,FZ,WCX,WCY,WCZ,ALFA,CALFA,SALFA
      DIMENSION TRAJ(NTRAJ),F9X9(41)
C
C  PICK OUT REQUIRED TRAJECTORY VARIABLES
      RLAT  = TRAJ(1)
      VX    = TRAJ(2)
      VY    = TRAJ(3)
      VZ    = TRAJ(4)
      FX    = TRAJ(5)
      FY    = TRAJ(6)
      FZ    = TRAJ(7)
      ALFA  = TRAJ(2) + HALFPI
C
C  COMPUTE 'SHARED' AND TEMPORARY VARIABLES
      CALFA = COS(ALFA)
      SALFA = SIN(ALFA)
      VE    = CALFA*VX - SALFA*VY
      VN    = SALFA*VX + CALFA*VY
      VG    = SQRT(VX**2 + VY**2)
      FE    = CALFA*FX - SALFA*FY
      FN    = SALFA*FX + CALFA*FY
      CLAT  = COS(RLAT)
      SLAT  = SIN(RLAT)
      TLAT  = SLAT / CLAT
      RCLAT = 1./CLAT
      OMEGN = OMEGA*CLAT
      OMEGZ = OMEGA*SLAT
      RHOE  =-VN*RREQ
      RHON  = VE*RREQ
      RHOZ  = VE*RREQ*TLAT
      WE    = RHOE
      WN    = RHON+OMEGN
      WZ    = RHOZ+OMEGZ
      WCX   = CALFA*RHOE + SALFA*WN
      WCY   =-SALFA*RHOE + CALFA*WN
      WCZ   = OMEGZ
      RKZ   = VZ*RREQ
C
C  EVALUATE THE TIME-DEPENDENT NON-ZERO ELEMENTS
C  IN THE FUNDAMENTAL ERROR DYNAMICS MATRIX
      F9X9 (1) = RHOZ*RCLAT
      F9X9 (2) = 2.*(OMEGN*VN+OMEGZ*VZ)+RHON*VN*RCLAT*RCLAT
      F9X9 (3) =-2.*OMEGN*VE-RHON*VE*RCLAT*RCLAT
      F9X9 (4) =-2.*OMEGZ*VE
      F9X9 (5) =-OMEGZ
      F9X9 (6) = WN*RHOZ*TLAT
      F9X9 (7) =-RHON*RREQ*RCLAT
      F9X9 (8) = RHOE*RREQ
      F9X9 (9) =-CK1
      F9X9(10) = RHOZ*RHOE*RHON*KKZ
      F9X9(11) = RHON*RHOZ-RHOE*XKZ
      F9X9(12) = 2.*GEE*RREQ-(RHON*RHON*RHOE*RHOE)-CK2
      F9X9(13) =-RHOE*RREQ
      F9X9(14) =-RHON*RREQ
      F9X9(15) =-RHOZ*RREQ
      F9X9(16) = RREQ*RCLAT
      F9X9(17) =-RHOE*TLAT-XKZ
      F9X9(18) =-2.*WZ
      F9X9(19) = 2.*WN
      F9X9(21) = TLAT*RREQ
      F9X9(23) = WZ+OMEGZ
      F9X9(24) =-XKZ
      F9X9(25) =-2.*RHOE
      F9X9(28) =-(WN+OMEGN)
      F9X9(29) = RHOE
      F9X9(30) = FZ
      F9X9(31) =-FN
      F9X9(32) =-WZ
      F9X9(33) = WN
      F9X9(34) =-FZ
      F9X9(35) = FE
      F9X9(36) = WZ
      F9X9(37) =-WE
      F9X9(38) = FN
      F9X9(39) =-FE
      F9X9(40) =-WN
      F9X9(41) = WE
      RETURN
C
C  ..............................................................
      ENTRY ERDYO
C
C  EVALUATE THE TIME-INDEPENDENT NON-ZERO ELEMENTS
C  IN THE FUNDAMENTAL ERROR DYNAMICS MATRIX
      RREQ     = 1./REQ
      F9X9(20) = RREQ
      F9X9(22) = RREQ
      F9X9(26) =-RREQ
      F9X9(27) = 1.
      RETURN
      END
```

```
 SUBROUTINE ESTIX       74/74    OPT=1                              FTN 4.8+564        11/03/82  15.33.3

                    SUBROUTINE ESTIX(IRUN,T,NF,NS,NKTJ,XF,XS,XTRAJ,NTR,PF)
              C
                    ENTRY ESTIXO
                    RETURN
                    END
```

```
SUBROUTINE FOGEN       74/74    OPT=1                          FTN 4.8+564        11/09/82  15.33

      SUBROUTINE FOGEN(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ,NZF,NZQ,F,QF)
C
      DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ),F(NZF),QF(NZQ)
      RETURN
      ENTRY FOGEND
      RETURN
      END
      SUBROUTINE HRZ(IRUN,T,NF,NS,NXTJ,EDX,TDX,TX,NTR,PF,
     *                  IMEAS,M,H,RVAR,ZRES)
C
      RETURN
      ENTRY HRZO
      RETURN
      END
```

```
                SUBROUTINE NUUNIT
          C
          C  NUUNIT CONVERTS STATISTICAL INPUT TO COMPUTATIONAL UNITS
          C
                COMMON /CORPS/DALTS,DGRAVE,DGRAVN,DGA
                COMMON /MATH/QTRPI,HALFPI,PI,TWOPI,RPD
                COMMON /NOISDS/GYNDS(3),ACNDS(3)
                COMMON /SIGO/DYNAMIC(10),GB(3),GF(6),GFF(3),GSF(3),GM(6),
               *                AB(3),ASF(3),AM(6),BARO,GDE,GND,GA,
               *                BASF,SPC,AL
                COMMON /SIGMAS/BASIGS,EDSIGS,DNSIGS,GASIGS
          C
          C
          C  FORM CONVERSION FACTORS FOR CHANGING INPUT DATA
          C  TO COMPUTATIONAL UNITS
          C
          C  *** RPD    = RADIANS PER DEGREE
          C  *** SPH    = SECONDS PER HOUR
          C  *** ACCPG  = FT/SEC/SEC PER SEE
          C  *** ACCPUG = FT/SEC/SEC PER MICRO SEE
          C  *** RPAS   = RADIANS PER ARC SECOND
          C  *** FPNM   = FEET PER NAUTICAL MILE
          C
                RPD     = HALFPI/90.
                SPH     = 3600.
                ACCPG   = 32.2
                ACCPUG  = 32.2/1000000.
                RPAS    = RPD/3600.
                FPNM    = 1852./.3048
          C
          C  CHANGE STATISTICAL INPUT DATA TO COMPUTATIONAL UNITS
          C
                DO 10 I=7,9
           10      DYNAMIC(I) = DYNAMIC(I)/1000.
                DO 20 I=1,3
                   GB(I)     = GB(I)*RPD/SPH
                   AB(I)     = AB(I)*ACCPUG
                   GYNDS(I) = ((GYNDS(I)*RPD/SPH)**2.)/SPH
                   GFF(I)    = GFF(I)*RPD/(SPH*ACCPG*ACCPG)
                   GSF(I)    = GSF(I)/1000000.
                   ASF(I)    = ASF(I)/1000000.
                   ACNDS(I) = ((ACNDS(I)*ACCPUG)**2.)/SPH
           20   CONTINUE
          C
                DO 30 I=1,6
                   GF(I) = GF(I)*RPD/(SPH*ACCPG)
                   GM(I) = GM(I)*RPAS
                   AM(I) = AM(I)*RPAS
           30   CONTINUE
          C
                DALTS  = DALTS*FPNM
                DGRAVE = DGRAVE*FPNM
                DGRAVN = DGRAVN*FPNM
                DGA    = DGA*FPNM
                GDE    = GDE*ACCPUG
                GND    = GND*ACCPUG
                GA     = GA*ACCPUG
```

```
                    EDSIGS = EDSIGS*ACCPUG
                    ONSIGS = ONSIGS*ACCPUG
                    GASIGS = GASIGS*ACCPUG
C
                    RETURN
                    END
```

```
SUBROUTINE SNOYS       74/74    OPT=1                              FTN 4.8+564         11/09/82  15.33

      SUBROUTINE SNOYS(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ)
C
C  USER-WRITTEN SUBROUTINE.
C  ADDS GAUSSIAN RANDOM SAMPLES TO SPECIFIED TRUTH STATES TO
C  SIMULATE THE ACCUMULATED EFFECT OF PROCESS DRIVING NOISE
C  ON THESE STATES OVER THE NOISE ACCUMULATION INTERVAL DT.
C
      COMMON /CORRS/DALTS,DGRAVE,DGRAVN,DGA
      COMMON /NOISDS/GYNDS(3),ACNDS(3)
      COMMON /SIGMAS/BASIGS,EDSIGS,DNSIGS,GASIGS
      DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ)
C
      DT     = T-TOLD
      VE     = XTRAJ(2)
      VN     = XTRAJ(3)
      VG     = SQRT(VE*VE+VN*VN)
C
      SIG11  = SQRT(DT*GYNDS(1))
      XS(11) = XS(11)+GAUSS(0.,SIG11)
      SIG12  = SQRT(DT*GYNDS(2))
      XS(12) = XS(12)+GAUSS(0.,SIG12)
      SIG13  = SQRT(DT*GYNDS(3))
      XS(13) = XS(13)+GAUSS(0.,SIG13)
      SIG32  = SQRT(DT*ACNDS(1))
      XS(32) = XS(32)+GAUSS(0.,SIG32)
      SIG33  = SQRT(DT*ACNDS(2))
      XS(33) = XS(33)+GAUSS(0.,SIG33)
      SIG34  = SQRT(DT*ACNDS(3))
      XS(34) = XS(34)+GAUSS(0.,SIG34)
      SIG44  = BASIGS*SQRT(1.-EXP(-2.*DT*VG/DALTS))
      XS(44) = XS(44)+GAUSS(0.,SIG44)
      SIG45  = EDSIGS*SQRT(1.-EXP(-2.*DT*VG/DGRAVE))
      XS(45) = XS(45)+GAUSS(0.,SIG45)
      SIG46  = DNSIGS*SQRT(1.-EXP(-2.*DT*VG/DGRAVN))
      XS(46) = XS(46)+GAUSS(0.,SIG46)
      SIG47  = GASIGS*SQRT(1.-EXP(-2.*DT*VG/DGA))
      XS(47) = XS(47)+GAUSS(0.,SIG47)
      TOLD   = T
      RETURN
C
C ****************************************************************
      ENTRY SNOYSO
      TOLD = T
      RETURN
      END
```

```
            SUBROUTINE STABLE
      C
      C  STABLE PRINTS A TABLE OF STATISTICAL INPUT DATA FOR THE TRUTH MODEL
      C
            COMMON /CORRS/DALTS,DGRAVE,DGRAVN,DGA
            COMMON /EARTH/OMEGA,REQ,ESQ,GEE
            COMMON /NOICOS/GYNDS(3),ACNDS(3)
            COMMON /SIGMAS/BASIGS,EDSIGS,DNSIGS,GASIGS
            COMMON /SIGD/DYNAMIC(10),GB(3),GF(6),GFF(3),GSF(3),GM(6),
           *              AB(3),AIF(3),AM(6),BARO,GDE,GND,GA,
           *              BASF,SPC,AL
      C
            DIMENSION DIRECT(3),SOURCE(19),AXIS(12),AXMIS(6),DYNAM(10)
            DATA AXIS/*X *,*X *,*Y *,*Y *,*Z *,*Z *,* INPUT(X)*,* SPIN(Y)*,
           ** SPIN(X)*,* INPUT(Y)*,* SPIN(Y)*,* INPUT(Z)*/
            DATA AXMIS/*ABT Y*,*ABT Z*,*ABT X*,*ABT Z*,*ABT X*,*ABT Y*/
            DATA DIRECT/*X *,*Y *,*Z */
            DATA DYNAM/*-COS(LT)FT*,* FEET*,* FEET*,* FT/SEC*,* FT/SEC*,
           ** FT/SEC*,* MRAD*,* MRAD*,* MRAD*,* FT/SEC**2*/
            DATA SOURCE/*EAST *,*LONGITUDE*,*NORTH*,*LATITUDE*,*ALTITUDE*,*
           * *,*EAST*,*VELOCITY*,*NORTH*,*VELOCITY*,*VERTICAL*,*VELOCITY*,
           * *EAST*,*ATTITUDE*,*NORTH*,*ATTITUDE*,*VERTICAL*,*ATTITUDE*/
      C
      C  WRITE TABLE OF TRUTH MODEL STATISTICAL INPUT DATA
            WRITE(6,100)
      C
            WRITE(6,200) 1,SOURCE(1),SOURCE(2),DYNAMIC(1)*REQ,DYNAM(1)
            WRITE(6,200) 2,SOURCE(3),SOURCE(4),DYNAMIC(2)*REQ,DYNAM(2)
            II=5
            DO 10 I=3,9
               N=II+1
               WRITE(6,200) I,SOURCE(II),SOURCE(N),DYNAMIC(I),DYNAM(I)
               II=II+2
       10   CONTINUE
      C
            WRITE(6,300) DYNAMIC(10),DYNAM(10)
      C
            DO 20 I=11,13
               N=I-10
               WRITE(6,400) I,DIRECT(N),GB(N),GYNDS(N)
       20   CONTINUE
      C
            DO 30 I=14,19
               N=I-13
               NN=N+6
               WRITE(6,500) I,AXIS(N),AXIS(NN),GF(N)
       30   CONTINUE
      C
            DO 40 I=20,22
               N=I-19
               WRITE(6,600) I,DIRECT(N),GFF(N)
       40   CONTINUE
      C
            DO 50 I=23,25
               N=I-22
               WRITE(6,700) I,DIRECT(N),GSF(N)
       50   CONTINUE
```

```
C
      DO 60 I=26,31
         N=I-25
         WRITE(6,500) I,AXIS(N),AXMIS(N),GM(N)
   60 CONTINUE
C
      DO 70 I=32,34
         N=I-31
         WRITE(6,900) I,DIRECT(N),AB(N),ACNOS(N)
   70 CONTINUE
C
      DO 80 I=35,37
         N=I-34
         WRITE(6,1000) I,DIRECT(N),ASF(N)
   80 CONTINUE
C
      DO 90 I=38,43
         N=I-37
         WRITE(6,1100) I,AXIS(N),AXMIS(N),AM(N)
   90 CONTINUE
C
      BANDS  = 2.*BASIGS*BASIGS/DALTS
      EDMNDS = 2.*EDSIGS*EDSIGS/DGRAVE
      DNNDS  = 2.*DNSIGS*DNSIGS/DGRAVN
      GANDS  = 2.*GASIGS*GASIGS/DGA
      WRITE(6,1200) BARO,BANDS,DALTS,GDE,EDMNDS,DGRAVE,GND,DNNDS,
     *                DGRAVN,GA,GANDS,DGA,BASF,SPC,AL
C
      RETURN
  100 FORMAT(*1*//T56,*TRUE ERROR STATISTICS*//T65,24(*.*),T93,
     *   *INPUT PARAMETERS *,22(*.*)//T5,*TRUTH*,T70,*NOISE*/
     *   T5,*STATE*,T65,*INITIAL ONE*,T93,*SPECTRAL*,T117,*CORRELATION*,
     *   /T5,*INDEX*,T13,*ERROR SOURCE*,T44,*ERROR MODEL*,
     *   T65,*SIGMA VALUE*,T90,*DENSITY*,T117,*PARAMETER*/)
  200 FORMAT(T7,I2,T13,2A9,T44,*DYNAMIC*,T62,G12.4,A10)
  300 FORMAT(T7,*10*,T13,*VERTICAL ACCELERATION*,T44,*DYNAMIC*,
     *   T62,G12.4,A10)
  400 FORMAT(T7,I2,T13,A2,T15,*GYRO DRIFT*,T44,*RANDOM WALK*,
     *   T62,G12.4,* DEG/HR*,T93,*(*,G10.2,
     *   * DEG/HR)**2/HR*)
  500 FORMAT(T7,I2,T13,A2,*GYRO G-SENS DRIFT,*,A9,T44,*RANDOM CONSTANT*,
     *   T62,G12.4,* DEG/HR/G*)
  600 FORMAT(T7,I2,T13,A2,*GYRO G*G-SENS DRIFT *,T24,T44,
     *   *RANDOM CONSTANT*,T62,G12.4,* DEG/HR/(G*G)*)
  700 FORMAT(T7,I2,T13,A2,T15,*GYRO SCALE FACTOR*,T44,*RANDOM CONSTANT*,
     *   T62,G12.4,* PPM*)
  800 FORMAT(T7,I2,T13,A2,*GYRO MISALIGNMENT *,A6,T44,*RANDOM CONSTANT*,
     *   T62,G12.4,* ARC SEC*)
  900 FORMAT(T7,I2,T13,A2,T15,*ACCELEROMETER BIAS*,T44,*RANDOM WALK*,
     *   T62,G12.4,* UGEE*,T93,*(*,G10.2,* UGEE)**2/HR*)
 1000 FORMAT(T7,I2,T13,A2,T15,*ACCELEROMETER SCALE FACTOR*,T44,
     *   *RANDOM CONSTANT*,T62,G12.4,* PPM*)
 1100 FORMAT(T7,I2,T13,A2,*ACCEL MISALIGNMENT *,A6,T44,
     *   *RANDOM CONSTANT*,T62,G12.4,* ARC SEC*)
 1200 FORMAT(T7,*44*,T13,*BARO ALTIMETER BIAS*,T44,
     *   *FIRST ORDER MARKOV*,T62,G12.4,* FEET*,T90,G10.2,
     *   * FEET**2/NM*,T112,G10.2,* NAUT MILES*/
```

```
      *      T7,*45*,T13,*E GRAVITY DEFLECTION*,T44,
      *   *FIRST ORDER MARKOV*,T62,G12.4,* UGEE*,T90,G10.2,
      *   * UGEE**2/NM*,T112,G10.2,* NAUT MILES*/
      *      T7,*46*,T13,*N GRAVITY DEFLECTION*,T44,
      *   *FIRST ORDER MARKOV*,T62,G12.4,* UGEE*,T90,G10.2,
      *   * UGEE**2/NM*,T112,G10.2,*NAUT MILES*/
      *      T7,*47*,T13,*GRAVITY ANOMLY*,T44,
      *   *FIRST ORDER MARKOV*,T62,G12.4,* UGEE*,T90,G10.2,
      *   * UGEE**2/NM*,T112,G10.2,* NAUT MILES*/
      *      T7,*48*,T13,*ALTIMETER SCALE FACTOR*,T44,
      *   *RANDOM CONSTANT*,T62,G12.4,* ********/
      *      T7,*49*,T13,*STATIC PRESSURE COEFFCIENT*,T44,
      *   *RANDOM CONSTANT*,T62,G12.4,* FT/(FT/SEC**2)*/
      *      T7,*50*,T13,*ALTIMETER LAG*,T44,
      *   *RANDOM CONSTANT*,T62,G12.4,*SECS*)
       END
```

```
            SUBROUTINE TRAJ(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ)
      C
      C   TRAJ CREATES AN EXTERNAL FLIGHT PROFILE  OF DURATION
      C   650 SECONDS.IT STARTS WITH A LEVEL FLIGHT WITH A SPEED
      C   OF 600MILES PER HOUR FOR 500 SECONDS.IT COMMENSES THE
      C   DIVE FOR 10 SECS WITH DOWNWARD VELOCITY OF 6000FT/MIN.
      C   AND LEVELS OF AT 1000FT.THIS TRAJECTORY WAS CREATED TO
      C   ENABLE A VEHICLE TO PERFORM A TERCOM UPDATE.
      C
            COMMON /MATH/QTRPI,HALFPI,PI,TWOPI,RPD
            COMMON/EARTH/OMEGA,REQ,ESQ,SEE
            COMMON/HGTH/HT
            DIMENSION XF(NF), XS(NS), XTRAJ(NXTJ)
      C
      C  SEGMENT ONE
            IF(T-T1)21,22,22
      21    VZ=0.
      C  INITIAL HEIGHT INFEET
            HT=11000.
      C  THE FOLLOWING EQUATIONS DESCRIBE A GREAT CIRCLE PATH
      C
                VHOR = GAMDOT*(REQ+HT)
                GAM = VHOR*(T-T0)/(REQ+HT)
                CTLAT = SQRT(COS(GAM)*COS(GAM)+((COS(TINCL)*SIN(GAM))**2))
                STLAT =  SIN(TINCL)*SIN(GAM)
                SALPHA = COS(TINCL)/CTLAT
            IF(GAM-G11)24,25,25
      25    IF(GAM-G12)26,26,24
      26    CALPHA=SQRT(1-SALPHA*SALPHA)
            GO TO 27
      24    CALPHA=-SQRT(1-SALPHA*SALPHA)
            GO TO 27
      C  SEGMENT TWO
      22    IF(T-T1)40,40,41
      40    GO TO 21
      41    IF(T-(T1+DELTA))42,42,43
      42    VZ=-50.*(1.-COS(OMEG*(T-T1)))
            GO TO 48
      43    IF(T-(T2-DELTA))44,44,45
      44    VZ=-100.
            GO TO 48
      45    IF(T-T2)46,46,29
      46    VZ=-50.*(1.-COS(OMEG*(T-T2)))
            GO TO 48
      48    HT = 11000.-100.*(T-T1)
                VHOR = GAMDOT*(REQ+HT)
                GAM = VHOR*(T-T0)/(REQ+HT)
                CTLAT = SQRT(COS(GAM)*COS(GAM)+((COS(TINCL)*SIN(GAM))**2))
                STLAT = SIN(TINCL)*SIN(GAM)
                SALPHA = COS(TINCL)/CTLAT
            IF(GAM-G11)30,31,31
      31    IF(GAM-G12)32,32,30
      32    CALPHA=SQRT(1-SALPHA*SALPHA)
            GO TO 27
      30    CALPHA=-SQRT(1-SALPHA*SALPHA)
      C  SEGMENT THREE
      29    VZ=0.
```

```
                 HT = 1000.
                 VHOR = GAMDOT*(REQ+HT)
                 GAM = VHOR*(T-T0)/(REQ+HT)
                 CTLAT = SQRT(COS(GAM)*COS(GAM)+((COS(TINCL)*SIN(GAM))**2))
                 STLAT = SIN(TINCL)*SIN(GAM)
                 SALPHA = COS(TINCL)/CTLAT
              IF(GAM-G11)33,34,34
      34      IF(GAM-G12)35,35,33
      35      CALPHA=SQRT(1-SALPHA*SALPHA)
              GO TO 27
      33      CALPHA=-SQRT(1-SALPHA*SALPHA)
      C
      C       ..........DEFINITIONS OF TRAJECTORY VARIABLES..........
      C
      C                    XTRAJ(1)     LATITUDE
      C                    XTRAJ(2)     X VELOCITY
      C                    XTRAJ(3)     Y VELOCITY
      C                    XTRAJ(4)     Z VELOCITY
      C                    XTRAJ(5)     X SPECIFIC FORCE
      C                    XTRAJ(6)     Y SPECIFIC FORCE
      C                    XTRAJ(7)     Z SPECIFIC FORCE
      C                    XTRAJ(8)     WANDER ANGLE, ALPHA
      C                    XTRAJ(9)     HEIGHT
      C
      C       WHERE, FOR ALPHA = -90 DEGREES, X-Y-Z POINT E-N-U RESPECTIVELY.
      C
      27      VN=VHOR*CALPHA
              VE = VHOR*SALPHA
              FE = -2*VHOR*WIE*STLAT*CALPHA
              FN = 2*VHOR*WIE*STLAT*SALPHA
              FZ = GEE-(VHOR*VHOR/(REQ+HT))-2*VHOR*WIE*COS(TINCL)
              XTRAJ(1) = ACOS(CTLAT)
              XTRAJ(2) = VE
              XTRAJ(3) = VN
              XTRAJ(4) = VZ
              XTRAJ(5) = FE
              XTRAJ(6) = FN
              XTRAJ(7) = FZ
              XTRAJ(9) = HT
              RETURN
              ENTRY TRAJO
      C  HORIZONTAL VELOCITY 600MILES/HOUR
                 VHOR = 1013.37
      C  INITIAL LATITUDE IN RADIANS
                 TLAT = 0.
      C  TIME OF SEGMENTS IN SECONDS
                 T0 = 0.
                 T1 = 500.
                 T2 = 600.
      C  GREAT CIRCLE PATH INCLINATION IN RADIANS
                 TINCL = .7953985
      C  INITIAL HEIGHT IN FEET
                 HT = 11000.
                 GAMDOT = VHOR/(REQ+HT)
                 G11 = -1.570797
                 G12 = 1.570797
                 G22 = 4.712391
```

```
                    WIE = .7292115147E-4
                 DELTA=.01
                 OMEG=PI/DELTA
           C  WANDER ANGLE IN RADIANS
                    XTRAJ(3) = -HALFPI
                 RETURN
                 END
```

```
              SUBROUTINE USRIN
        C
        C  USER-WRITTEN SUBROUTINE
        C  THIS ROUTINE SPECIFIES CONSTANTS AND READS
        C  IN CONTROL AND STATISTICAL DATA
        C
              COMMON /CORRS/DALTS,DGRAVE,DGRAVN,DGA
              COMMON /EARTH/OMEGA,REQ,ESQ,GEE
              COMMON /CNTRL/LFDBK,LINTOO,KKSGO
              COMMON /MATH/QTRPI,HALFPI,PI,TWOPI,RPD
              COMMON /NOISDS/GYNDS(3),ACNDS(3)
              COMMON /SIGC/DYNAMIC(10),GB(3),GF(6),GFF(3),GSF(3),GM(5),
             *            AB(3),ASF(3),AM(5),BARO,GDE,GND,GA,
             *            BASF,SPC,AL
              COMMON /SIGMAS/BASIGS,EDSIGS,DNSIGS,GASIGS
              COMMON /VDAMP/CK1,CK2,CK3
        C
              LOGICAL LFDBK,LINTOO
        C
        C
              NAMELIST/CONTRL/LFDBK,LINTOO,KKSGO
              NAMELIST/SIGCS/DYNAMIC,GB,GF,GFF,GSF,GM,AB,ASF,AM,
             *                BARO, GDE,GND,GA,BASF,SPC,AL
              NAMELIST/STATS/DALTS,DGRAVE,DGRAVN,DGA,GYNDS,ACNDS,BASIGS,EDSIGS,
             *                DNSIGS,GASIGS
        C
        C  SET EARTH-RELATED CONSTANTS
              OMEGA = .7292115147E-4
              REQ   = 2.092564E7
        C     ESQ   = 0.006694317778
              ESQ   = 0.0
              GEE   = 32.2
        C
        C  SET VERTICAL LOOP DAMPING CONSTANTS
              CK1 = 3.0E-2
              CK2 = 3.0307E-4
              CK3 = 1.0E-6
        C
        C  COMPUTE MATH CONSTANTS
              PI      = ACOS(-1.)
              QTRPI  = 0.25*PI
              HALFPI = 0.50*PI
              TWOPI  = 2.00*PI
              RPD    = PI/180.
        C
        C  READ AND ECHO STATISTICAL DATA
              READ (5,SIGCS)
              READ (5,STATS)
              WRITE(6,SIGCS)
              WRITE(6,100)
              WRITE(6,STATS)
              WRITE(6,300)
        C
        C  READ AND ECHO INITIALIZATION CONTROL PARAMETERS.

           NOTE THE FOLLOWING ABOUT CONTROL PARAMETER KKSGO:
        C    KKSGO = 1   TRUTH STATES INITIALIZED IN XSDOTO BY OVERWRITING
```

```
      C                     XS0 INPUT WITH ONE-SIGMA VALUES FROM NAMELIST *SIGOS*.
      C     KXSGO = 2       TRUTH STATES INITIALIZED BY MONTE CARLO PROCESS
      C                     USING ONE-SIGMA VALUES FROM NAMELIST *SIGOS*.
      C     OTHERWISE       ALL TRUTH STATES INITIALIZED FROM
      C                     XS0 INPUT (5TH ITEM IN TABLE 4-2 OF SCFE MANUAL).
            READ (5,CONTRL)
            WRITE(6,CONTRL)
            WRITE(6,400)
      C
      C  PRINT TABLES OF STATISTICAL INPUT
            CALL STABLE
      C
      C  CONVERT INPUT DATA TO COMPUTATIONAL UNITS AND ECHO IT ONCE MORE
            CALL NUUNIT
            WRITE(6,SIGOS)
            WRITE(6,STATS)
      C
            RETURN
        100 FORMAT(*0*,5X,*NOTE: THE SIGOS NAMELIST CONTAINS SIGMA *
           .*VALUES FOR INITIALIZATION OF TRUTH MODEL STATES (XS0) UNDER *
           .*CONTROL OF KXSGO AS FOLLOWS:*//T13,*KXSGO*T37,*ACTION*
           ./T15,*1*,T23,*XS0 SET TO ONE-SIGMA VALUES FROM SIGOS*,
           ./T15,*2*,T23,*XS0 MONTE CARLOED.  EACH OF NS SAMPLES IS GAUSSIAN*
           ./T23,*WITH MEAN ZERO AND STANDARD DEVIATION PROVIDED BY *
           .      *SIGOS DATA.*
           ./T13,*OTHER*,T23,*XS0 ASSIGNED PER DATA CARD INPUT*
           .///T13,*SIGOS DATA IS ALSO USED TO CONSTRUCT STRENGTH *
           .        *VALUES IN FOGEN FOR FILTER COVARIANCE TUNING.* )
        300 FORMAT(*0*,5X*NOTE: THE STATS NAMELIST CONTAINS SIGMA VALUES, NOIS
           .E SPECTRAL DENSITIES */T13,*AND CORRELATION PARAMETERS FOR INJECTI
           .NG NOISE IN THE TRUTH MODEL.*)
        400 FORMAT(*0*,5X*NOTE: THE CONTRL NAMELIST CONTAINS PARAMETERS THAT G
           .OVERN PROGRAM FLOW.*)
            END
      SUBROUTINE XFDOT(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ,NTR,PF,XDOT)
C
C  USER-WRITTEN SUBROUTINE WHICH COMPUTES THE FILTER DERIVATIVES
C
      DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ),PF(NTR),XDOT(NF)
      RETURN
      ENTRY XFDOTO
      XDOT(1) = 0.
      RETURN
      END
      SUBROUTINE XSDOT(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ,XDOT)
C
C  USER-WRITTEN SUBROUTINE WHICH COMPUTES THE TRUTH DERIVATIVES
C
      COMMON /CORRS/DALTS,DGRAVE,DGRAVN,DGA
      COMMON /CNTRL/LFDBK,LINTGO,KXSGO
      COMMON /SHAPE/VX,VY,VZ,VG,FX,FY,FZ,WCX,WCY,WCZ,ALFA,CALFA,SALFA
      COMMON /SIGO/DYNAMIC(10),GB(3),GF(5),GFF(3),GSF(3),GM(6),
     *           AB(3),ASF(3),AM(5),BARO,GDE,GMD,GA,
     *           BASF,SPC,AL
      COMMON /VDAMP/CK1,CK2,CK3
      COMMON/HGTH/HT
      COMMON/MATH/QTRPI,HALFPI,PI,TWOPI,RPD
C
      DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ),XDOT(NS),F9X9(41)
      IF(T-T1)13,13,14
13    DTB=0.
      GO TO 21
14    IF(T-(T1+DELTA))15,15,16
15    DTB=100.*3.2E0*.33333*(1.-COS(OMEG*(T-T1)))
      GO TO 21
16    IF(T-(T2-DELTA))17,17,18
17    DTB=200.*3.2E0*.33333
      GO TO 21
18    IF(T-T2)19,19,25
19    DTB=100.*3.2E0*.33333*(1.-COS(OMEG*(T-T2)))
      GO TO 21
25    DTB=0.
C
C
C  CALL TRAJ TO SET FLIGHT PATH
21     CALL TRAJ(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ)
C
C
C  CALL SUBROUTINE ERDY TO OBTAIN TIME-DEPENDENT ELEMENTS
C  OF THE FUNDAMENTAL NAVIGATION ERROR DYNAMICS MATRIX AND TO
C  OBTAIN SENSED ACCELERATION AND COMMANDED PLATFORM ANGULAR VELOCITY
       CALL ERDY(NXTJ,XTRAJ,F9X9)
C
C  SET ELEMENTS OF CNP.  *N* IS LL-ENZ WHILE *P* IS THE
C  WANDER AZIMUTH FRAME DENOTED LL-XYZ.
       CNP11 = CALFA
       CNP12 =-SALFA
       CNP21 = SALFA
       CNP22 = CALFA
C
C  COMPUTE TRUTH DERIVATIVES :
C  BASIC NINE ERROR STATES :
      XDOT (1) = F9X9 (1)*XS(2)+F9X9 (7)*XS(3)+F9X9(16)*XS(4)
      XDOT (2) = F9X9 (8)*XS(3)+F9X9(22)*XS(5)
      XDOT (3) = F9X9 (9)*XS(3)+XS(6)+CK1*XS(44)+(CK1*HT)*XS(48)
     *             +CK1*VG*VG*XS(49)-CK1*VZ*XS(50)+CK1*DTB
      XDOT (4) = F9X9 (2)*XS(2)+F9X9(10)*XS(3)+F9X9(17)*XS(4)
     *             +F9X9(23)*XS(5)+F9X9(24)*XS(6)+F9X9(34)*XS(9)
     *             +F9X9(38)*XS(7)+CNP11*XS(32)  +CNP12*XS(33)
     *             +CNP11*FX*XS(35)+CNP12*FY*XS(36)-CNP11*FZ*XS(37)
     *             +CNP11*FY*XS(39)+CNP12*FZ*XS(40)+CNP21*FX*XS(41)+XS(45)
```

```
      XDOT (5) = F9X9 (3)*XS(2)+F9X9(11)*XS(3)+F9X9(18)*XS(4)
     *           +F9X9(24)*XS(5)+F9X9(29)*XS(6)+F9X9(30)*XS(7)
     *           +F9X9(39)*XS(9)+CNP21*XS(32)  +CNP22*XS(33)
     *           +CNP21*FX*XS(35)+CNP11*FY*XS(36)+CNP12*FZ*XS(37)
     *           +CNP21*FY*XS(39)+CNP11*FZ*XS(40)-CNP22*FX*XS(41)+XS(46)
      XDOT (6) = F9X9 (4)*XS(2)+F9X9(12)*XS(3)+F9X9(19)*XS(4)
     *           +F9X9(25)*XS(5)+F9X9(31)*XS(7)+F9X9(35)*XS(8)
     *           -XS(10)*XS(34)+FZ*XS(37)-FY*XS(42)+FX*XS(43)
     *           +CK2*XS(44)+CK2*HT*XS(48)+XS(47)+CK2*VG*VG*XS(49)
     *           -CK2*VZ*XS(50)+CK2*DTB
      XDOT (7) = F9X9(13)*XS(3)+F9X9(26)*XS(5)+F9X9(36)*XS(8)
     *           +F9X9(40)*XS(9)+CNP11*XS(11)+CNP12*XS(12)
     *           +CNP11*FX*XS(14)+CNP11*FY*XS(15)+CNP12*FX*XS(16)
     *           +CNP12*FY*XS(17)+CNP11*FX*FY*XS(20)+CNP12*FX*FY*XS(21)
     *           +CNP11*WCX*XS(23)+CNP12*WCY*XS(24)+CNP11*WCZ*XS(26)
     *           -CNP11*WCY*XS(27)+CNP21*WCZ*XS(28)+CNP12*WCX*XS(29)
      XDOT (8) = F9X9 (5)*XS(2)+F9X9(14)*XS(3)+F9X9(20)*XS(4)
     *           +F9X9(32)*XS(7)+F9X9(41)*XS(9)+CNP21*XS(11)
     *           +CNP22*XS(12)+CNP21*FX*XS(14)+CNP21*FY*XS(15)
     *           +CNP11*FX*XS(16)+CNP11*FY*XS(17)+CNP21*FX*FY*XS(20)
     *           +CNP11*FX*FY*XS(21)+CNP21*WCX*XS(23)+CNP11*WCY*XS(24)
     *           +CNP21*WCZ*XS(25)+CNP12*WCY*XS(27)-CNP11*WCZ*XS(28)
     *           +CNP11*WCX*XS(29)
      XDOT (9) = F9X9(6)*XS(2)+F9X9(15)*XS(3)+F9X9(21)*XS(4)
     *           +F9X9(33)*XS(7)+F9X9(37) XS(9)+XS(13)+FY*XS(18)
     *           +FZ*XS(19)+FY*FZ*XS(22)+WCZ*XS(25)+WCY*XS(30)
     *           -WCX*XS(31)
C
C  VERTICAL ACCELERATION ERROR STATE ;
      XDOT(10) = XS(3)*CK3-CK3*XS(44)+CK3*VZ*XS(50)
     *            -CK3*VG*VG*XS(49)-CK3*HT*XS(48)-CK3*DTB
C
C  BARO ALTIMETER AND GRAVITY MISALIGNMENT ERRORS ;
      XDOT(44) =-VG*XS(44)/DALTS
      XDOT(45) =-VG*XS(45)/DGRAVE
      XDOT(46) =-VG*XS(46)/DGRAVN
      XDOT(47) =-VG*XS(47)/DGA
C
*
*
*
      RETURN
C
C  ****************************************************************
      ENTRY XSDOTC
      T1=500.
      T2=600.
      T3=610.
      T4=660.
      DELTA=.01
      OMEG=PI/DELTA
C
C  CALL ERDYD TO SET THE TIME-INDEPENDENT ELEMENTS OF
C  THE FUNDAMENTAL NAVIGATION ERROR DYNAMICS MATRIX
      CALL ERDYD(NXTJ,XTRAJ,F9X9)
C
C  SET DERIVATIVES FOR RANDOM CONSTANT
```

```
C  AND RANDOM WALK ERROR STATES
      DO 10  I=11,43
   10 XDOT(I) = 0.
      DO 20  I=48,50
   20 XDOT(I)=0.
C
      IF (KXSGO.NE.1 .AND. KXSGO.NE.2) RETURN
C
C  SET DYNAMIC TRUTH STATES
      DO 30 I=1,10
   30    XS(I) = DYNAMIC(I)
C  SET RANDOM CONSTANT TRUTH STATES
      XS(14) = GF(1)
      XS(15) = GF(2)
      XS(16) = GF(3)
      XS(17) = GF(4)
      XS(18) = GF(5)
      XS(19) = GF(6)
      XS(20) = GFF(1)
      XS(21) = GFF(2)
      XS(22) = GFF(3)
      XS(23) = GSF(1)
      XS(24) = GSF(2)
      XS(25) = GSF(3)
      XS(26) = GM(1)
      XS(27) = GM(2)
      XS(28) = GM(3)
      XS(29) = GM(4)
      XS(30) = GM(5)
      XS(31) = GM(6)
      XS(35) = ASF(1)
      XS(36) = ASF(2)
      XS(37) = ASF(3)
      XS(38) = AM(1)
      XS(39) = AM(2)
      XS(40) = AM(3)
      XS(41) = AM(4)
      XS(42) = AM(5)
      XS(43) = AM(6)
      XS(48)=BASF
      XS(49)=SPC
      XS(50)=AL
C
C  INITIALIZE RANDOM WALK TRUTH STATES
      XS(11) = GB(1)
      XS(12) = GB(2)
      XS(13) = GB(3)
      XS(32) = AB(1)
      XS(33) = AB(2)
      XS(34) = AB(3)
C
C  INITIALIZE FIRST ORDER MARKOV TRUTH STATES
      XS(44) = BARO
      XS(45) = GDE
      XS(46) = GND
      XS(47) = GA
C
```

```
SUBROUTINE XSDOT       74/74    OPT=1                                  FTN 4.8+564         11/08/82  15.33

                   IF (KXSGO.EQ.1) RETURN
             C
                   DO 40 I=1,49
            40        XS(I) = GAUSS(0.,XS(I))
                   RETURN
                   END
```

```
RUN DATE AND TIME  :   11/08/82  15.34.42.
```

Vita

Asad Riaz was born on 16 October 1951 in Karachi, Pakistan. He was graduated from the Cadet College in Hasanabdal, Pakistan, in 1968. He was then enrolled in the Pakistan Air Force and sent to the College of Aeronautical Engineering, Pakistan, and was awarded a B.E. degree in Avionics in October 1972. During his various assignments in the Air Force, he was responsible for the maintenance and testing of the weapon systems. In May 1981, he was selected to join the Graduate Guidance and Control Program under the Electrical Engineering Department of the U.S. Air Force Institute of Technology. He is married to Zebinda, and their first child, a daughter, was born during the course of the graduate program at AFIT.

Permanent Address: 89-B, Gulberg II
Hali Road
Lahore, Pakistan

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/82D-55	2. GOVT ACCESSION NO. AD-A124 882	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* INVESTIGATION OF A THIRD-ORDER BARO-DAMPED VERTICAL CHANNEL OF INS		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) ASAD RIAZ Squadron Leader, PAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS ASD/ENACN Wright-Patterson AFB OH 45433		12. REPORT DATE December 1982
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)*
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT *(of this Report)* Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*		
18. SUPPLEMENTARY NOTES Approved for public release: IAW AFR 190-17. LYNN E. WOLAVER Dean for Rese[illegible] Development Air Force Inst. [illegible] Wright-Patters[illegible] 4 JAN 1		
19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)* INS, Vertical Channel, Baro-Inertial		
20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)* The optimization of the three gains of a third-order baro-inertial vertical channel has been formulated as a stochastic optimal control problem, with the objective of minimizing the mean squared altitude error due to the noise induced altitude error and a disturbance of known magnitude. For a vehicle carrying out a TERCOM-update immediately following a vertical descent, and being subjected to a disturbance input to the vertical channel,		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

Item 20 Cont'd

optimum gains are presented and the performance is analyzed through a simulated flight in a Monte Carlo analysis. Performance comparisons between the optimized gains and the classical gains are also presented. The results show a significant performance improvement over the classical gains for a vehicle carrying out the TERCOM-update.

END

FILMED

3-83

DTIC